Toward Unsupervised, Multi-Object Discovery in Large-Scale Image Collections

Huy V. Vo[1,2,3], Patrick Pérez[3], and Jean Ponce[1,2]

[1] INRIA, Paris, France
[2] Département d'informatique de l'ENS, ENS, CNRS, PSL University, Paris, France
[3] Valeo.ai

1 Regularized OSD (rOSD)

We have presented in the paper a new version of the OSD formulation [7] with added constraints based on the structure of our region proposals. Concretely, we propose to solve the optimization problem:

$$\max_{x,e} S(x,e) = \sum_{i=1}^{n} \sum_{j \in N(i)} e_{ij} x_i^T S_{ij} x_j, \text{ s.t.} \forall i \begin{cases} \sum_{k=1}^{p} x_i^k \leq \nu, \\ \sum_{k \in G_{ig}} x_i^k \leq 1, \text{ for all groups } g \\ \sum_{j \neq i} e_{ij} \leq \tau. \end{cases} \quad (1)$$

We solve this problem with an iterative block-coordinate ascent algorithm similar to OSD. Its iterations are illustrated in Algorithm 1.

Note that the output of Algorithm 1 depends on the order in which the variables x_i are processed in its first *for* loop. In our implementation, we use a *different* random permutation of $(1, ..., n)$ in each iteration of the optimization. For each experiment, we run rOSD several times and report the average performance of all runs as the final performance.

2 Large-Scale Object Discovery Algorithm

We summarize in Algorithm 2 our proposed large-scale algorithm for object discovery.

3 Experimental Results

3.1 Results with the Ensemble Method from [7]

Vo *et al.* [7] use an ensemble method (EM) to combine several solutions before post processing to stabilize and improve the final performance of OSD. We investigate the influence of this procedure on the performance of OSD and rOSD with our proposals, and present the result in Tables 1 and 2. We use VGG16 features in these experiments. It can be seen that the effect of EM is mixed for

Algorithm 1: Block coordinate ascent algorithm for rOSD.

Result: A solution to rOSD.
Input: G_i, ν, τ, S_{ij}, number n of images.
Initialization: $x_i = \mathbb{1}_p \ \forall i$, $e_{ij} = 1 \ \forall i \neq j$.
for $i = 1$ *to* n **do**
  Compute the vector R containing the scores of regions in image i.
  $R \longleftarrow \sum_{j \neq i}^{n} (e_{ij} S_{ij} + e_{ji} S_{ji}^T) x_j$.
  $I \longleftarrow \emptyset$.
  for $g = 1;\ g \leq L_i$ **do**
    Find the region g^* with highest score $R(g^*)$ in the group G_{ig}.
    $I \longleftarrow I \cup \{g^*\}$.
  end
  Choose ν regions in I with highest scores in R, assign their corresponding variables to 1. Assign the variables of other regions to 0.
end
for $i = 1$ *to* n **do**
  Compute the indices j_1 to j_τ of the τ largest scalars $x_i^T S_{ij} x_j$ $(1 \leq j \leq n)$.
  $e_i \longleftarrow 0$.
  for $t = 1;\ t \leq \tau$ **do**
    $e_{ij_t} \longleftarrow 1$.
  end
end

the tested datasets. It generally harms the performance on VOC_all and VOC12 and improves the performance on VOC_6x2 while its effect on OD is unclear. We have therefore chosen to omit EM in the experiments of the main body of the paper.

3.2 Full Results with both VGG16 and VGG19 Features

We present in Tables 3, 4 and 5 our full results in colocalization and object discovery with features from both VGG16 and VGG19. It can be seen that, with VGG16 features, rOSD still significantly outperforms OSD on the two large datasets and fares comparably to OSD on the smaller two. It is also noticeable that rOSD significantly outperforms Wei *et al.* in both colocalization and single-object discovery on all datasets when VGG16 features are used.

3.3 Multi-Object Experiments

For a fair comparison to OSD and Wei *et al.* [8] in multi-object discovery, we have fixed the number of objects retained in each image by all methods to 5 in the paper. We have also modified the method of Wei *et al.* such that 5 bounding boxes around the 5 largest clusters of positive pixels in their *indicator matrix* are returned as objects. For OSD and rOSD, we run the corresponding optimization then apply the following post processing on each image: all ν retained regions are ranked in descending order using the score proposed in [7] (Eq. 12 in Sec. 2.6

Algorithm 2: Large-scale object discovery algorithm.

Input: Dataset D of n images, memory limit M, number of partition k, image neighborhood size N, ν^*, τ.
Partition D into random k parts D_1, ..., D_k, each has roughly $\lfloor n/k \rfloor$ images.
Compute the maximum number of positive entries in the score matrices in each parts: $K_1 \longleftarrow M/(N * \lfloor n/k \rfloor)$.
Compute the maximum number of positive entries in the score matrices in the whole dataset: $K_2 \longleftarrow M/(n * N)$.
for $i = 1$ *to* k **do**
  Compute score matrices for image pairs in D_i with K_1 positive entries.
  Run proxy OSD on D_i with $\nu = K_2$.
  Each image in D_i has a new set of region proposals which are those retained by OSD.
end
Compute score matrices between pairs of images in D with K_2 positive entries.
Run OSD on the whole dataset D with $\nu = \nu^*$.

Table 1: Influence of the ensemble method of Vo *et al.* on the colocalization performance of OSD and rOSD with our proposals

Method		OD	VOC_6x2	VOC_all	VOC12
Ours (OSD)	w/o EM	89.0 ± 0.6	73.6 ± 0.6	44.7 ± 0.3	49.0 ± 0.2
Ours (OSD)	w/ EM	88.2 ± 0.2	**75.3 ± 0.2**	44.7 ± 0.1	48.7 ± 0.1
Ours (rOSD)	w/o EM	89.0 ± 0.5	73.3 ± 0.5	**45.8 ± 0.3**	**49.7 ± 0.1**
Ours (rOSD)	w/ EM	**89.2 ± 0.3**	74.5 ± 0.2	45.5 ± 0.1	**49.7 ± 0.2**

therein), which is solely based on their similarity to the retained regions in the image's neighbors; We then iteratively discard all proposals having an IoU score greater than some threshold with higher-ranked regions; Among remaining regions, we return the 5 highest ranked as retrieved objects. Since this procedure can eliminate all but a few regions if the regions highly overlap, we choose a large value of ν (50) and a large value of IoU threshold (0.7) in our experiments to guarantee that we have *exactly* 5 objects. This is, however, just a design choice and one can choose to retain fewer or more regions. We have conducted experiments with the number of retrieved objects varied in the interval $[2, 10]$

Table 2: Influence of the ensemble method of Vo *et al.* on the single-object discovery performance of OSD and rOSD with our proposals

Method		OD	VOC_6x2	VOC_all	VOC12
Ours (OSD)	w/o EM	87.8 ± 0.4	69.2 ± 0.5	48.7 ± 0.3	51.3 ± 0.2
Ours (OSD)	w/ EM	87.5 ± 0.3	70.9 ± 0.3	48.6 ± 0.1	50.7 ± 0.1
Ours (rOSD)	w/o EM	87.6 ± 0.3	71.1 ± 0.8	**49.2 ± 0.2**	**52.1 ± 0.1**
Ours (rOSD)	w/ EM	**88.7 ± 0.3**	**71.9 ± 0.4**	48.7 ± 0.1	52.0 ± 0.1

Table 3: Single-object colocalization performance of our approach compared to the state of the art. Note that Wei *et al.* [8] outperform our method on VOC_all and VOC12 with VGG19 features in this case, but the situation is clearly reversed in the much more difficult single-object discovery setting, as demonstrated in Table 4

Method	Features	OD	VOc_6x2	VOC_all	VOC12
Cho *et al.* [2]	WHO	84.2	67.6	37.6	-
Vo *et al.* [7]	WHO	87.1 ± 0.5	71.2 ± 0.6	39.5 ± 0.1	-
Li *et al.* [5]	VGG16	-	-	40.0	41.9
Wei *et al.* [8]	VGG16	86.9	66.2	44.7	47.6
Ours (OSD)	VGG16	89.0 ± 0.6	73.6 ± 0.6	44.7 ± 0.3	49.0 ± 0.2
Ours (rOSD)	VGG16	89.0 ± 0.5	73.3 ± 0.5	45.8 ± 0.3	49.7 ± 0.1
Li *et al.* [5]	VGG19	-	-	41.9	45.6
Wei *et al.* [8]	VGG19	87.9	67.7	**48.7**	**51.1**
Ours (OSD)	VGG19	**90.3 ± 0.3**	75.3 ± 0.7	45.6 ± 0.3	47.8 ± 0.2
Ours (rOSD)	VGG19	90.2 ± 0.3	**76.1 ± 0.7**	46.7 ± 0.2	49.2 ± 0.1

Table 4: Single-object discovery performance in the mixed setting on the datasets with our proposals compared to the state of the art

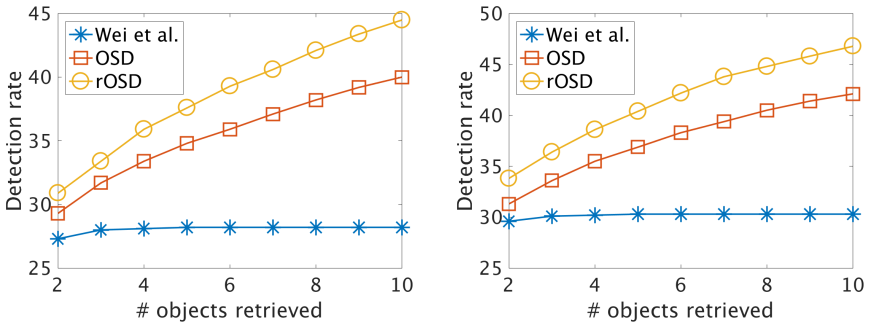
Method	Features	OD	VOC_6x2	VOC_all	VOC12
Cho *et al.* [2]	WHO	82.2	55.9	37.6	-
Vo *et al.* [7]	WHO	82.3 ± 0.3	62.5 ± 0.6	40.7 ± 0.2	-
Wei *et al.* [8]	VGG16	73.5	66.2	41.9	45.0
Ours (OSD)	VGG16	87.8 ± 0.4	69.2 ± 0.5	48.7 ± 0.3	51.3 ± 0.2
Ours (rOSD)	VGG16	87.6 ± 0.3	71.1 ± 0.8	49.2 ± 0.2	**52.1 ± 0.1**
Wei *et al.* [8]	VGG19	75.0	54.0	43.4	46.3
Ours (OSD)	VGG19	89.1 ± 0.4	71.9 ± 0.7	47.9 ± 0.3	49.2 ± 0.2
Ours (rOSD)	VGG19	**89.2 ± 0.4**	**72.5 ± 0.5**	**49.3 ± 0.2**	51.2 ± 0.2

and observed that rOSD always yields better performance than OSD and [8] regardless of the number of objects retrieved (Fig. 1).

Images may of course contain fewer than 5 objects. In such cases, OSD and rOSD usually return overlapping boxes around the actual objects (Fig. 3 in the paper). We can eliminate these overlapping boxes and obtain better qualitative results by using smaller ν and IoU threshold. We have conducted preliminary experiments with $\nu = 25$ in the optimization of OSD and rOSD and $IoU = 0.3$ for suppression threshold in the post processing and show qualitative results in Fig. 2. It can be seen that rOSD is now able to return bounding boxes around objects without many overlapping regions. It is also observed that rOSD fares much better than OSD in localizing multiple objects. We also compare the quantitative performance of rOSD, OSD and [8] in Table 6. For [8], we take as before the bounding boxes around the largest clusters of pixels in the *indicator matrix* of each image. The number of clusters in this case is chosen to be the number of objects returned by rOSD in the same image. The results show that rOSD again yields by far the best performance. It is also noticeable that while using smaller

Table 5: Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets

Method	Features	Colocalization		Discovery	
		VOC_all	VOC12	VOC_all	VOC12
Vo *et al.* [7]	WHO	40.7 ± 0.1	-	30.7 ± 0.1	-
Wei *et al.* [8]	VGG16	38.3	40.4	25.8	28.2
Ours (OSD)	VGG16	45.9 ± 0.1	48.1 ± 0.0	34.9 ± 0.1	37.6 ± 0.0
Ours (rOSD)	VGG16	48.5 ± 0.1	50.7 ± 0.1	37.2 ± 0.1	**40.8 ± 0.1**
Wei *et al.* [8]	VGG19	43.3	45.5	28.1	30.3
Ours (OSD)	VGG19	46.8 ± 0.1	47.9 ± 0.0	34.8 ± 0.0	36.9 ± 0.0
Ours (rOSD)	VGG19	**49.4 ± 0.1**	**51.5 ± 0.1**	**37.6 ± 0.1**	40.4 ± 0.1



(a) VOC_all (b) VOC12

Fig. 1: Multi-object discovery performance of rOSD compared to OSD and [8] when varying the maximum number of returned objects.

values of ν and the IoU threshold slightly deteriorates the performance of rOSD, it makes the performance of OSD drop significantly (compare Tables 5 and 6). This is due to the fact that OSD returns many highly overlapping regions and most of them are eliminated by our procedure. On the other hand, rOSD returns more diverse regions and consequently more regions are retained. In practice, we observe that OSD returns on average 1.47 (respectively 1.52) regions while rOSD returns 3.62 (respectively 3.63) on VOC_all (respectively VOC12). Note, however, that rOSD still outperforms OSD and [8] even when the latter are allowed to retain exactly 5 regions.

3.4 Evaluating the Graph Computed by OSD

Following [2], we evaluate the local graph structure obtained by rOSD using the CorRet measure, defined as the average percentage of returned image neighbors that belong to the same (ground-truth) class as the image itself. As a baseline, we consider the local graph induced by the sets of nearest neighbors $N(i)$ computed

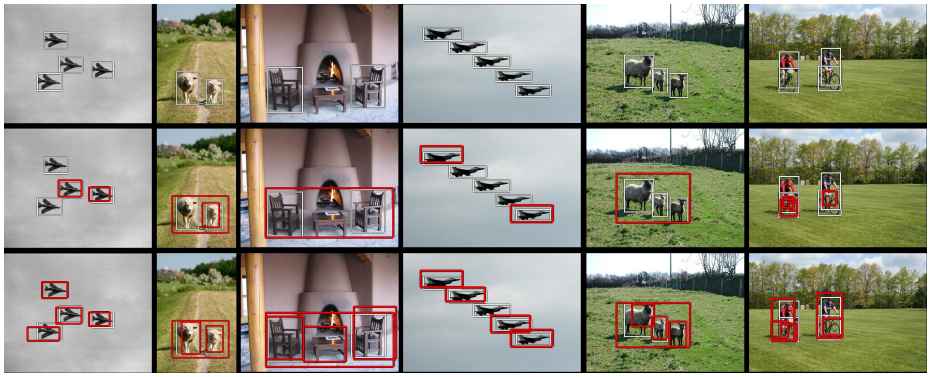


Fig. 2: Multi-object discovery results. In each column, from top to bottom: original image, image with predictions of OSD, image with predictions of rOSD. White boxes are ground truth objects and red ones are our predictions. There are *at most* 5 predictions per image.

Table 6: Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets when using smaller values of ν (25) and IoU (0.3) threshold

Method	Features	Colocalization		Discovery	
		VOC_all	VOC12	VOC_all	VOC12
Wei *et al.* [8]	VGG19	43.1	45.3	27.8	30.0
Ours (OSD)	VGG19	39.6 ± 0.1	41.6 ± 0.1	29.0 ± 0.1	31.3 ± 0.1
Ours (rOSD)	VGG19	**47.3 ± 0.1**	**49.3 ± 0.1**	**36.7 ± 0.1**	**39.2 ± 0.1**

from the fully connected layer *fc6* of the CNN that are used in the same experiment. Table 7 shows the CorRet of local graphs obtained when running rOSD (OSD) on VOC_all and VOC12 and large-scale rOSD (OSD) on COCO_20k in the mixed setting. It can be seen that the local image graphs returned by our methods have higher CorRet than the baseline.

3.5 Results on Images of ImageNet Classes not in the Training Set of the Feature Extractors

Though trained for classifying 1000 object classes of ImageNet, features from convolutional layers of VGGs have shown to be generic: They have been used

Table 7: Quality of the returned local image graph as measured by CorRet

Dataset	VOC_all	VOC12	COCO_20k
Baseline	50.7	56.4	36.8
Ours (OSD)	**60.1 ± 0.1**	**63.2 ± 0.0**	**39.8 ± 0.0**
Ours (rOSD)	59.8 ± 0.1	63.0 ± 0.0	39.4 ± 0.0

Table 8: Colocalization and single-object discovery performance of rOSD compared to OSD, Li *et al.* [5] and Wei *et al.* [8] on 6 *held-out* ImageNet classes

Method	Features	Colocalization	Discovery
Li *et al.*[4] [5]	VGG16	48.3	-
Wei *et al.* [8]	VGG16	<u>74.3</u>	61.2
Ours (OSD)	VGG16	61.5 ± 0.3	60.3 ± 0.3
Ours (rOSD)	VGG16	63.0 ± 0.7	<u>61.6 ± 0.4</u>
Li *et al.* [5]	VGG19	51.6	-
Wei *et al.* [8]	VGG19	**74.8**	**63.2**
Ours (OSD)	VGG19	61.3 ± 0.5	59.2 ± 0.7
Ours (rOSD)	VGG19	63.7 ± 0.3	59.4 ± 0.5

for various tasks, including unsupervised object discovery. Li *et al.* [5] and Wei *et al.* [8] have shown that CNN features generalize well beyond the classes in ILSVRC2012 by testing on 6 held-out classes on ImageNet (*chipmunk*, *racoon*, *rhinoceros*, *rake*, *stoat* and *wheelchair*). We have also tested our method on these classes. Since ImageNet has been under maintenance, we could not download all the official images in the six classes. For preliminary experiments, we have instead downloaded the images using their public URLs (provided on the ImageNet website), eliminated corrupted images, randomly chosen up to 200 images per class and run our experiments on these images. We have compared rOSD, OSD, [5] and [8] in this setting (Table 8). Although rOSD performs significantly better than [5] in colocalization tasks, it is as before significantly outperformed by [8] there. In object discovery, rOSD performs slightly better than [8] for VGG16 features, but significantly worse for VGG19 features. Understanding this discrepancy observed in preliminary experiments is part of our plans for future work.

4 More Visualizations

4.1 Overlapping Regions Returned by OSD and rOSD

The most important advantage of rOSD over OSD is that the former returns more diverse regions than the former does. We visualize the regions returned by OSD and rOSD in colocalization experiments with $\nu = 5$ in Fig. 3.

4.2 Persistence

We use persistence [1,3,4,6,9] to find robust local maxima of the global saliency map s_g in our work. Considering s_g as a 2D image and each location in it as a pixel, we associate with each pixel a cluster (the 4-neighborhood connected component of pixels that contains it), together with both a "birth" (its own saliency) and "death time" (the highest value for which one of the pixels in its

[4] Numbers for [5] are taken from [8].

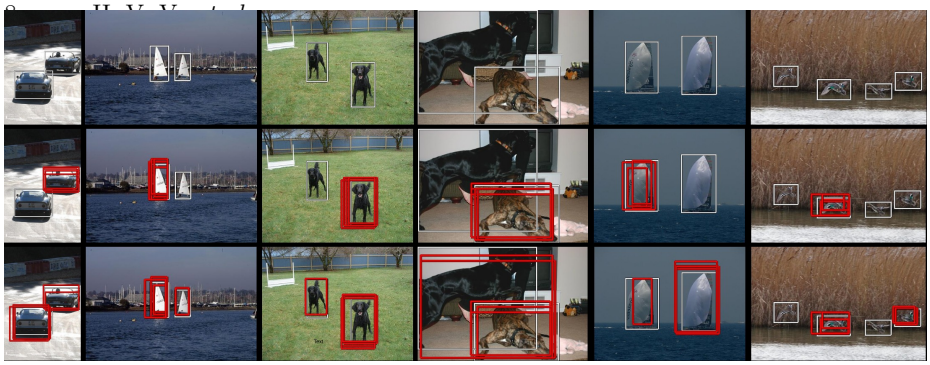


Fig. 3: Regions returned by OSD and rOSD. In each column, from top to bottom: original image, image with regions returned by OSD, image with regions returned by rOSD.

cluster also belongs to the cluster of a pixel with higher saliency, or, if no such location exists, the lowest saliency value in the map). The persistence of a pixel is defined as the difference between its birth and death times. Figure 4 illustrates persistence for the 1D case.

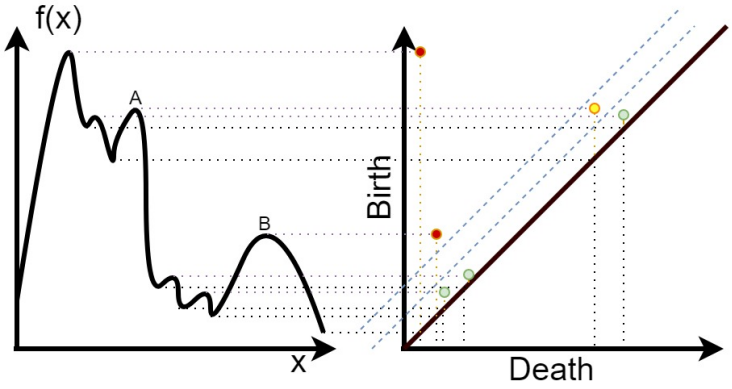


Fig. 4: An illustration of persistence in the 1D case. Left: A 1D function. Right: Its persistence diagram. Points above the diagonal correspond to its local maxima and the vertical distance from these points to the diagonal is their persistence. Local maxima with higher persistence are more robust: B is more robust than A although $f(A) > f(B)$. Given a chosen persistence threshold (shown by dash lines in blue), points with persistence higher than some threshold are selected as robust local maxima. The black horizontal dotted lines show birth and death time of the local maxima of f.

References

1. Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in riemannian manifolds. Journal of the ACM **60**(6), 41:1–41:38 (2013) 7
2. Cho, M., Kwak, S., Schmid, C., Ponce, J.: Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 4, 5
3. Edelsbrunner, H., Harer, J.: Computational Topology: An Introduction. AMS Press (2009) 7
4. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete and Computational Geometry (2002) 7
5. Li, Y., Liu, L., Shen, C., Hengel, A.: Image co-localization by mimicking a good detector's confidence score distribution. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016) 4, 7
6. Oudot, S.: Persistence Theory: From Quiver Representations to Data Analysis. AMS Surveys and Monographs (2015) 7
7. Vo, H.V., Han, K., Cho, M., Pérez, P., Bach, F., LeCun, Y., Ponce, J.: Unsupervised image matching and object discovery as optimization. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 2, 4, 5
8. Wei, X.S., Zhang, C.L., Wu, J., Shen, C., Zhou, Z.H.: Unsupervised object discovery and co-localization by deep descriptor transforming. Pattern Recognition (PR) **88** (2019) 2, 4, 5, 6, 7
9. Zomorodian, A., Carlsson, G.: Computing persistent homology. Discrete and Computational Geometry (2005) 7