BING3D: Fast Spatio-Temporal Proposals for Action Localization

Ella Gati
University of Amsterdam

John G. M. Schavemaker
TNO

Jan C. van Gemert
Delft University of Technology

Abstract

The goal of this work is realistic action localization in video with the aid of spatio-temporal proposals. Current proposal generation methods are computationally demanding and are not practical for large-scale datasets. The main contribution of this work is a novel and fast alternative. Our method uses spatio-temporal gradient computations, a generalization of BING to the temporal domainleading to BING3D. The method is orders of magnitude faster than current methods and performs on par or above the localization accuracy of current proposals on the UCF sports and MSR-II datasets. Furthermore, due to our efficiency, we are the first to report action localization results on the large and challenging UCF 101 dataset. Another contribution of this work is our Apenheul case study, where we created and tested our proposals performance on a novel and challenging dataset. The Apenheul dataset is large-scale, as it contains full high definition videos, featuring gorillas in a natural environment, with uncontrolled background, lighting conditions and quality.

1. Introduction

Action localization (1) deals with unraveling *when*, *where* and *what* happens in a video. In this work we propose a method for action localization using spatio-temporal proposals, which is fast and achieves state-of-the-art results. The naive approach to action localization is using a sliding sub-volume, which is the 3D extension of the sliding window approach for static images. While effective for static images [15], when dealing with videos sliding window approaches become computationally intractable even for modest sized videos.

More recent methods for action localization [10, 20, 30] are proposals based. This is inspired by successful object-proposals methods in static images [1, 4, 19, 29]. Proposal based methods first reduce the search space to a small set of spatio-temporal tubes, with high likelihood to contain an action. Compared to sliding-subvolume approaches, such as [12, 24, 26], proposals for action localization are more efficient and allow using bigger datasets. Another advantage of proposal based methods is that the small number of proposals that has to be classified makes it possible to use more computationally expensive features and more advanced classifiers, that would be impractical otherwise, to achieve state-of-the-art localization accuracy.

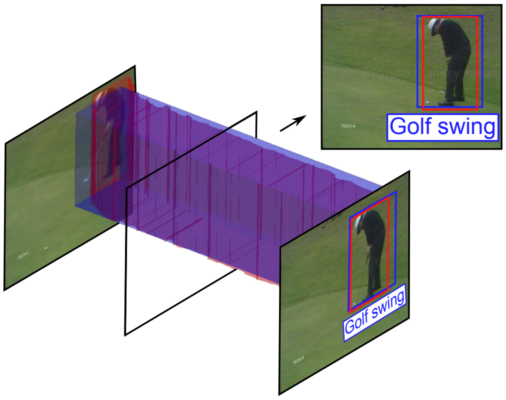


Figure 1. Action localization aims to find where, when and what action is taking place in a video. The red tubelet is the ground truth, the blue cuboid is our best proposal. The action label indicates what action is taking place.

Current action proposal algorithms are based on dense trajectories [30] or use video segmentation [10, 20] to generate the action proposals. Segmentation is computational expensive, and takes several minutes for a modest video of 720x400 video of 55 frames [20, 35] and can take days for a realistic full HD video. The computational demands of segmentation based action proposals are not practical for large-scale video processing. This is the main motivation for creating a fast large-scale action localization method.

In this work, we present BING3D, a generalization of BING [4] from image to video for high speed 3D proposals, in which we use spatio-temporal video gradients instead of video segmentation. We chose BING because of its impressive speed and small number of quality object proposals. The strength of BING's efficiency lies in simple gradient features and an approximation method for fast proposal selection. We generalize to the temporal domain by adding temporal features, and a spatio-temporal approximation method leading to BING3D.

BING3D is orders of magnitude faster than current methods and performs on par or above the localization accuracy of current proposals on benchmark datasets.

section 2 gives a short review of the related research in the field. The method is described and explained in section 3. We present the experimental setup, as well as experiments results and analysis in section 4. Finally we conclude our work in section 5.

2. Related Work

Several action localization methods apply an action classifier directly on the video. Examples include sliding 3D subvolume methods like spatio-temporal template matching [24], a 3D boosting cascade [12] and spatio-temporal deformable 3D parts [26]. Other methods maximize a temporal classification path of 2D boxes through static frames [27, 28] or search for the optimal classification result with a branch and bound scheme [36]. The benefit is that these methods do not require an intermediate representation and directly apply a classifier to densely sampled parts of the video. The disadvantage of such methods, however, is that they have to perform the same dense sampling for each individual action class separately. Due to the computational complexity of the sampling, this is impractical for larger numbers of action classes. Instead, we use spatio-temporal proposals to first generate a small set of bounding-box tubes that are likely to contain any type of action.

Current spatio-temporal proposals are inspired by 2D object proposals in static images. A version of objectness [1] is extened to video [2], selective search [29] led to tubelets from motion [10] and randomized Prim [19] was generalized to a spatio-temporal variant [20]. Several 2D object proposal methods and their 3D generalizations are based on a super-pixel segmentation pre-processing step [1, 2, 10, 14, 19, 20, 29] which we argue is computationally too demanding for large scale video processing. Other 2D proposal methods such as edge boxes [38] use edge-detection and BING [4] uses gradient computation as pre-processing steps. Since gradients are the fastest to compute we propose a 3D extension of BING for large-scale spatio-temporal action proposals. To avoid the expensive pre-processing step altogether, we also propose a method of generating proposals without any pre-processing. This second method generates proposals from the local features as required later on by the action classifier.

Local features provide a solid base for action recognition and action localization. Points are sampled at salient spatio-temporal locations [6, 17], densely [25, 34] or along dense trajectories [31, 33]. The points are represented by powerful local descriptors [18, 13, 5] that are robust to modest changes in motion and in appearance. Robustness to camera motion is either directly modeled from the video [9, 33] or dealt with at the feature level by the MBH descriptor [5, 31]. After aggregating local descriptors in a global video representation such as VLAD [9] or Fisher [21, 22, 33] they are input to a classifier like SVM. Due to the excellent performance of dense trajectory features [31, 33] in action localization [10], we adopt them as our feature representation throughout this paper.

3. Method

The generation of action proposals is done using BING3D, our extension of the BING [4] algorithm from static images to videos. BING stands for 'BInariazed Normalized Gradient', as it is based on image gradient as its basic features. Image derivatives, as well as their three-dimensional extension for videos, are simple features that can be computed efficiently. It has been shown that objects tend to have well-defined object boundaries [4], that are captured correctly by the spatial derivatives magnitude. Adding the temporal derivative to the gradient is imperative to capture the temporal extent of an action.

3.1. NG3D

We use normalized video gradients (referred to as NG3D) as the base to our features. The gradient of video v is defined by the partial derivatives of each dimension $|\nabla v| = |(v_x, v_y, v_z)^T|$, where v_x, v_y, v_z are the partial derivative of the x, y, z axes respectively. The partial derivatives are efficiently computed by convolving the video v with a 1D mask [-1 0 1], which is an approximation of the Gaussian derivative, in each dimension separately. For each pixel the gradient magnitude is computed and then clipped at 255 to fit the value in a byte, as $min(|v_x| + |v_y| + |v_z|, 255)$. The final feature vector is the L_1 normalized, concatenated gradient magnitudes of a pixel block. The shape of the pixel block is 8x8 spatially, so it fits in a single int64 variable, which allow for easy use of bitwise operations, and we vary the temporal depth of the feature D resulting in a $8 \times 8 \times D$ block. In section 4 we evaluate the performance when varying the temporal depth D.

Figure 2 illustrates the NG3D features. The top row is showing a sequence of random frames from one of the training videos. The red boxes are random boxes of non-action, while the green boxes cover a *Running* action. The bottom boxes illustrate the spatio-temporal NG3D features of the boxes drawn on top. The action is described with $D = 4$ temporal layers on NG3D feature, while random blocks from the same video do not display a similar pattern illustrating that the NG3D feature can be used for discriminating actions from non-actions.

In order to generate diverse proposals in terms of width, height and length, we first resize our videos to a set of predefined scales (1/2, 1/4, 1/8, 1/16, 1/32), using trilinear interpolation.

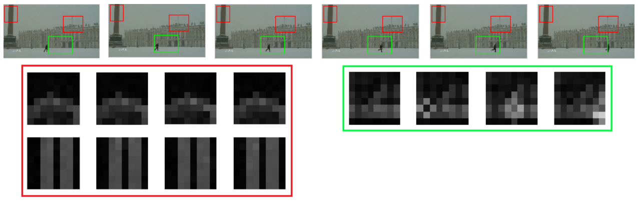


Figure 2. Visualization of 3D Normalized Gradients (NG3D). Top: The red boxes are on non-action parts in the video, the green box covers a *Running* action. Bottom: visualisation of the spatio-temporal NG3D features in the red and green boxes from the top in 8×8 spatial resolution and $D = 4$ temporal frames. The action is clearly described with NG3D feature, while random blocks from the same video do not display a similar pattern illustrating that the NG3D feature can be used for discriminating actions from non-actions.

3.2. BING3D

To compute BING3D, we learn a classifier model, compute its approximation and then binarize the NG3D features to what we call BING3D. The computed features and approximated model are used to compute proposal scores.

Learning a classifier model The positive samples in the train set are approximations of the ground truth tracks. Each track is enlarged to a cuboid and then resized with different scales. Cuboids that overlap more than a threshold (0.25) with the ground truth cuboid are used as positive samples. The negative samples are random cusboids that do ot overlap with any gt track. We use linear SVM to learn model w.

Approximate model Efficient proposal classification is achieved by approximating the SVM model w in a binary embedding [4, 8] which allows fast bitwise operations in the evaluation. The learned model $w \in \mathbb{R}^{8\times 8\times D}$ is approximated by a set of binary basis vectors $\mathbf{a} \in \{-1, 1\}^{8\times 8}$ and their coefficients $\beta \in \mathbb{R}$. The approximation becomes

$$w \approx \sum_{i=1}^{D}\sum_{j=1}^{N_w} \beta_{ij}\mathbf{a}_{ij}. \qquad (1)$$

In section 4 we evaluate the quality of the approximation with different number of components N_w. Pseudo code for computing the binary embedding is given in algorithm 1.

Generating BING3D features In addition to the approximation of the model, we also approximate the normed gradient values using the top N_g binary bits of the BYTE values. Thus, each dimension of the NG3D feature g_l can be approximated by N_g binarized normed gradient features as:

$$g_l = \sum_{k=1}^{N_g} 2^{8-k}\mathbf{b}_{k,l} \qquad (2)$$

where $l = (i, x, y, z)$ is the scale and location of the feature. The $8 \times 8 \times D$ patches of approximated gradient are the BING3D features. As with the approximation of w, we approximate each temporal slice independently. We use the fast algorithm proposed in [4], and presented in algorithm 2 to compute the 8×8 feature for each of the D temporal slices. Thanks to the cumulative relation between adjacent BING3D features and their last rows, we can avoid looping over the 8×8 region, by using BITWISE SHIFT and BITWISE OR operations.

Algorithm 1 Binary approximation of w

```
Input: $\mathbf{w}, N_w, D$
Output: $\{\{\beta_{ij}\}_{j=1}^{N_w}\}_{i=1}^{D}, \{\{\mathbf{a}_{ij}\}_{j=1}^{N_w}\}_{i=1}^{D}$
  for $i = 1$ to $D$ do
    $\varepsilon = \mathbf{w}_i$
    for $j = 1$ to $N_w$ do
      $\mathbf{a}_{ij} = \text{sign}(\varepsilon)$
      $\beta_{ij} = \langle \mathbf{a}_{ij}, \varepsilon\rangle / ||\mathbf{a}_{ij}||^2$
      $\varepsilon \leftarrow \varepsilon - \beta_{ij}\mathbf{a}_{ij}$
    end for
  end for
```

Algorithm 2 BING [4] algorithm to compute BING features for $W \times H$ positions

```
Input: binary normed gradient map $b_{W\times H}$
Output: BING feature matrix $\mathbf{b}_{W\times H}$
  Initialize: $\mathbf{b}_{W\times H} = 0, \mathbf{r}_{W\times H} = 0$
  for each position $(x, y)$ in scan-line order do
    $\mathbf{r}_{x,y} = (\mathbf{r}_{x-1,y} \ll 1) \quad | \quad b_{x,y}$
    $\mathbf{b}_{x,y} = (\mathbf{b}_{x,y-1} \ll 8) \quad | \quad \mathbf{r}_{x,y}$
  end for
```

Proposals Generation The proposal generation process involves computing an approximated classifier score (or 'proposal score') s_l for each scale and location in the video and then choosing only the top scored proposals.

The approximated classifier score is defined as

$$s_l = \langle w, g_l\rangle \qquad (3)$$

and can be efficiently tested using:

$$s_l \approx \sum_{i=1}^{D}\sum_{j=1}^{N_w}\beta_{ij}\sum_{k=1}^{N_g} 2^{8-k}(2\langle \mathbf{a}_{ij}^{+}, \mathbf{b}_{k,l}\rangle - |\mathbf{b}_{k,l}|) \qquad (4)$$

We use non-maximum suppression to reduce the number of proposals according to their proposal score.

3.3. Action Localization

We use the state-of-art descriptors computed along improved dense trajectory [33]. To represent a proposal, we aggregate all the visual words corresponding to the trajectories that fall inside of it. For training, we use a one-vs-rest linear SVM classifier.

4. Experiments

4.1. Datasets

We evaluate on three diverse datasets for action localization: **UCF Sports, UCF 101** and **MSR-II**. UCF Sports consists of 150 videos extracted from sports broadcasts of varying resolution; it is trimmed to contain a single action in all frames. UCF101 is collected from YouTube and has 101 action categories where 24 of them contain localization annotations, corresponding to 3,204 videos. All UCF101 videos contain exactly one action[1], most of them (74.6%) are trimmed to fit the action. In contrast, the MSR-II Action dataset consists of 203 actions in 54 videos where each video has multiple actions of three classes. The actions are performed in a crowded environment with ample background movement. The MSR-II videos are relatively long, on average 763 frames, and the videos are untrimmed.

4.2. Experimental Setup

For all experiments in this section we use a train-test split and state results obtained on the test set. For UCF Sports and UCF 101 we use the standard split, and for MSR-II a random split of 50% train and 50% test videos. Since UCF-sports and UCF 101 are trimmed, BING3D outputs full length proposals for them. Both in BING3D and in the localization training, we set the positive samples threshold to 0.25 in all experiments. We used liblinear [7] everywhere SVM is used, and the SVM parameter is set using cross validation. We used default parameters in the extraction of the improved dense trajectories. For the Fisher encoding, we always reduced descriptors' dimensionality to half, as suggested in [32].

In the experiments and evaluation of the algorithm we used three benchmark action localization datasets, namely UCF Sports, UCF 101 and MSR-II

We used different methods to quantify the performance of our algorithms. For the proposals quality evaluation we used the ABO, MABO and Best Overlap recall measures, as explained in more details next. The action localization is evaluated using average precision and AUC.

The proposal quality of a proposal P with a ground truth tube G is evaluated with spatio-temporal tube overlap measured as the average "intersection-over-union" score for 2D boxes for all frames where there is either a ground truth box or a proposal box. More formally, for a video V of F frames, a tube of bounding boxes is given by $(B_1, B_2, ...B_F)$, where $B_f = \emptyset$, if there is no action i in frame f, ϕ is the set of frames where at least one of G_f, P_f is not empty. The localization score between G and P is $L(G,P) = \frac{1}{|\phi|}\sum_{f\in\phi}\frac{G_f \cap P_f}{G_f \cup P_f}$.

The Average Best Overlap (ABO) score is computed by averaging the localization score of the best proposal for each ground truth action. The Mean Average Best Overlap (MABO) is the mean of the per class ABO score. The recall is the percentage of ground truth actions with best overlap score over a threshold. It is is worth mentioning that although other papers often use 0.2 as the threshold, we chose to use stricter criteria, thus unless stated otherwise we report recall with a 0.5 threshold.

The localization performance is measured in means of average precision (AP) and mean average precision (mAP). To compute average precision, the proposals are sorted according to their classification score. A proposal is considered relevant if its label is predicted correctly and its overlap score with the ground truth tubelet is over a threshold. We present plots of AP and mAP scores for different overlap thresholds. For comparability with previous works , we also provide AUC plot, computed as in [16].

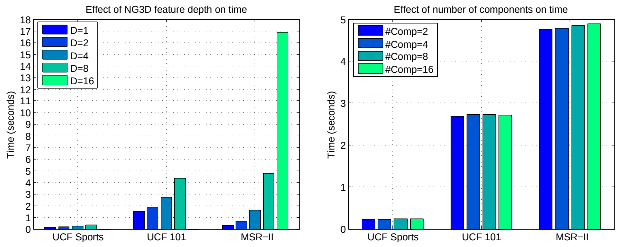
4.3. Experiments

Effect of NG3D feature depth (D). We vary the temporal NG3D feature depth $D \in \{1, 2, 4, 8, 16\}$ while keeping $N_w = 4$ fixed, see 1. In 3 (left) we report the average time per video in seconds where higher D values are slower. Next, we show the effect on the recall in 4 (left). The feature depth does not matter much for UCF-Sports and UCF 101. Even disregarding the temporal scale, $D = 1$, works well which is due to the trimmed nature of these datasets. For untrimmed MSR-II, where temporal localization is required, the best performance is obtained by higher D, which illustrates the need for temporal modeling in untrimmed videos.

Effect of model approximation (N_w). We vary $N_w \in \{2, 4, 8, 16\}$ while clamping D to the best value (4 for UCF-Sports, UCF-101, 8 for MSR-II). In 3 (right) we report the average time per video in seconds, showing that N_w has barely any effect on the computation time. The effect on recall is illustrated in 4 (right). The approximation quality does not effect accuracy for trimmed UCF-Sports and UCF 101 where even $N_w = 2$ components works well. For untrimmed MSR-II more than 2 components are needed, and $N_w = 16$ components is too much, which may reinterpret N_w as a regularization parameter.

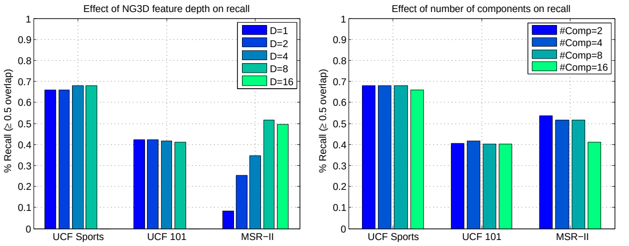
Cross-dataset model transfer Recently it was suggested that the supervision in the original BING is not crucial to

[1] We used the first annotated "person" in the XML file.



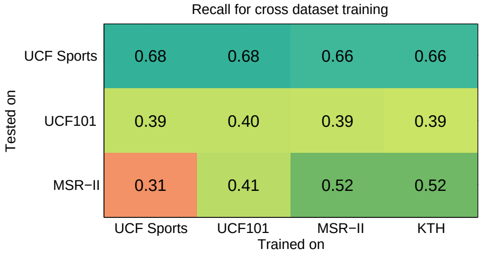
Effect of D Effect of N_w

Figure 3. Evaluating BING3D parameters D (left) and N_w (right) on computation time (s). The feature depth has a strong impact on the generation time.



Effect of D Effect of N_w

Figure 4. Evaluating BING3D parameters D (left) and N_w (right) on recall. The untrimmed MSR-II dataset is the most sensitive to parameter variations, illustrating the need for temporal modeling.

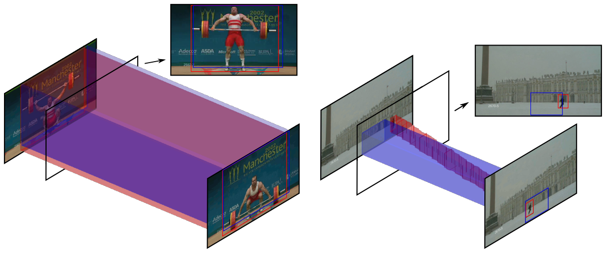

Recall for cross dataset training

Tested on \ Trained on	UCF Sports	UCF101	MSR−II	KTH
UCF Sports	0.68	0.68	0.66	0.66
UCF101	0.39	0.40	0.39	0.39
MSR−II	0.31	0.41	0.52	0.52

Figure 5. Cross dataset training a model on set A, and applying it on set B. Note the robustness on UCF-Sports and UCF101. The untrimmed MSR-II set is sensitive to model variations.

its success [37]. To test how this affects BING3D we evaluate the quality of the learned model w by training on one dataset, and evaluation on another dataset. For training, we include the spatio-temporal annotations of the KTH dataset [11], KTH is commonly used as a train set for MSR-II [10]. We show cross-dataset results in 5. For UCF Sports and UCF 101 the results are similar for all models. For MSR-II however, the model learned on the untrimmed MSR-II and KTH sets outperforms models trained on the trimmed datasets. We conclude that for trimmed videos the model has limited influence, yet, for untrimmed videos a model trained on untrimmed data is essential.

Qualitative analysis To get a better understanding of the strengths and weaknesses of BING3D, we analyze success and failure cases for each dataset. We visualize below the ground truth tracks with highest and lowest best overlap score. In all the figures the blue cuboid illustrates the proposal and the red one the ground truth. The overlap score is stated under each figure.



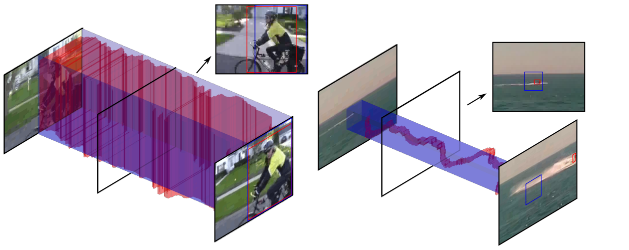
Overlap: 0.87 Overlap: 0.12

Figure 6. UCF Sports: visualization of best overlap proposals with highest and lowest overlap score.

The highest scored proposal for UCF Sports is from the *Lifting* class (figure 6 left). This class is characterized by cuboid ground truth annotations which makes it easier on BING3D to generate quality proposals. The lowest scored proposal (figure 6 right) is from the *running* class. Here we can see the weak point of generating only cuboids and not tubelets. Even though the proposal captures almost all of the action range (which can be seen by the fact that most of the ground truth tubelet is inside the proposal cuboid), the overlap score is low, because per frame there is a big difference in the bounding boxes sizes between the proposal and the ground truth.

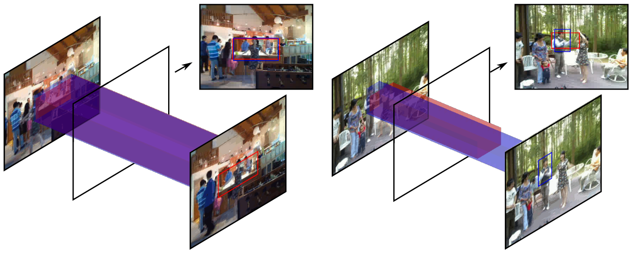
Figure 7 shows proposals for UCF 101. On the left, *Biking* action has Large bounding boxes that fit nicely in a cuboid, thus yielding high scored best proposal. On the right we encounter again the disadvantage of generating only cuboid proposals. Whenever an action contains large movements within the frame, the overlap scores are dropping. There are a few other ground truth tubelets with low overlap scores that were not visualized because they are too short (up to 20 frames), thus making the visualization unclear. Since we treated UCF 101 as a trimmed dataset, all proposals were generated with full video length and therefore for the few untrimmed videos, we get low overlap scores.

For MSR-II the big challenge is the temporal localization. The highest scored proposal is demonstrating impressive success, from a video with length of 907 frames, the temporal localization is only 4% off (126 common frames between the proposal and the ground truth, out of shared length of 131 frames, when the length of the ground truth tubelet is 129 frames). Encouraging results are that even for the lowest scored proposal (figure 8 right) the temporal localization is relatively good. 21 out of 32 frames are shared. The bad performance in this case might be again



Overlap: 0.81 Overlap: 0.05

Figure 7. UCF 101: visualization of best overlap proposals with highest and lowest overlap score.



Overlap: 0.84 Overlap: 0.29

Figure 8. MSR-II: visualization of best overlap proposals with highest and lowest overlap score.

	Computation time (s)		
	Pre-processing	Generation	Total
Prim3D	840	38	878
Tubelets	185	59	244
BING3D	**1**	**0.6**	**2**

Table 1. Computation times for pre-processing, proposal generation, and their combined total on a 400x720 video of 55 frames with 12,852 trajectories. Note the speedup of our proposals.

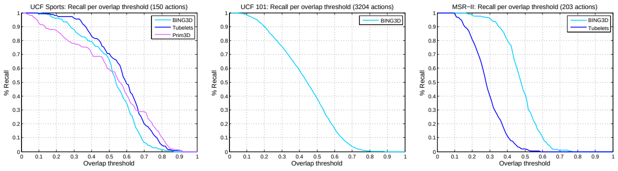
due to the short ground truth track. With average length of 320 frames per action, BING3D learns to generate longer proposal cuboids, thus failing to fit the outlier ground truth track temporally.

Versus state of the art In this section compare BING3D versus other action localization methods. The methods we compare to are the *Tubelets* method by Jain et al. [10] and *Prim3D* by Oneata et al. [20], for both of which we got the raw proposals, and computed all the evaluation metrics ourselves, so to have a fair comparison.

First of all, we compare the computation time of BING3D versus other methods. The strongest point of BING3D is its fast speed, orders of magnitude faster than other methods, as can be seen in table 1. We compare the processing time for one video from the UCF Sports dataset, for which we have timing results from the other methods. Our timing was measured on a single core, 2.93 Ghz Intel Xeon processor.

	ABO	MABO	Recall	#Proposals
UCF Sports				
Prim3D	51.83	50.89	57.79	3,000
Tubelets	**63.41**	**62.71**	**78.72**	1,642
BING3D	51.84	51.76	66.00	**300**
UCF 101				
BING3D	**43.10**	**42.80**	**38.17**	**1,700**
MSR-II				
Tubelets	34.88	34.81	2.96	**4,218**
BING3D	**47.56**	**47.54**	**41.38**	14,500

Table 2. Summary of our BING3D method performance, compared with other methods when available. While our performance is lower than Tubelets for UCF Sports, we still outperform Prim3D in all metrics. Note that we still perform well with 5 times less proposals. We significantly outperform Tubelets on MSR-II.

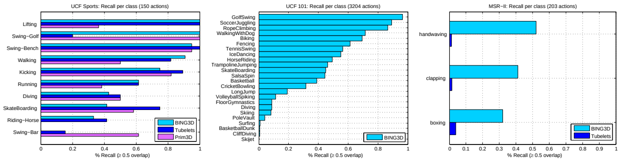


UCF UCF 101 MSR-II

Figure 9. Recall per threshold results for three benchmarks, comparing to other methods when applicable.

Next, we compare the performance with three evaluation metrics (ABO, MABO and recall), for the three benchmarks. We also state the number of proposals each method generated. Note that the number of proposals generated by BING3D is significantly lower. For UCF Sports our performance is lower than that of Tubelets, but we still outperform Prim3D in all metrics. Note that we still perform well with 10 to 15 times less proposals. UCF 101 does not have any previously reported results to compare to, and on MSR-II we significantly outperform Tubelets with about half the number of proposals. It is also important to remember that since BING3D outputs cuboids and not tubelets as the other methods, its performance is bounded.

Figure 9 shows the recall for different overlap thresholds on all datasets. As mentioned before we can see that BING3D is dominated by Tubelets for UCF Sports. We can also see that even though BING3D performs better than Prim3D for low thresholds (up to 0.5), it actually degrades for higher thresholds. Note that for the far more challenging dataset of MSR-II, where the additional temporal segmentation and the presence of multiple actions per video enlarges the search space a lot, BING3D still manages to maintain a relatively low number of proposals, while achieving high performance (over 98% for a 0.2 threshold, and over 41% for a 0.5 threshold). Figure 10 states the recall per class.



UCF UCF 101 MSR-II

Figure 10. Per class recall results for three benchmarks, comparing to other methods when applicable.

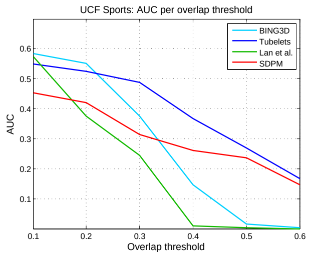


Figure 11. UCF Sports, action localization results.

4.4. Action Localization

We experimented with different settings for action localization, our results show that a combination of all the IDT features performs best, aggregated by Fisher vectors with K=128 Gaussian components, normalized using power normalization followed by l_2 normalization as was done Perronnin by *et al.* in [23].

Versus state of the art After the exhaustive parameter evaluation we chose the best parameters for the experiments on UCF 101 and MSR-II, these are *allBefore* features, $K = 128$, power normalization and no LCE. We present here the results, comparing to previous work when possible.

Figure 11 shows the area under the ROC curve for varying overlap thresholds for UCF Sports, comparing BING3D with *Tubelets* [10] from Jain et al., SDPM [26] from Tian et al, and work of Lan et al. [16]. For the lower thresholds (up to 0.4) BING3D outperforms the other methods, but it degrades fast for higher thresholds. This is a consequence of the proposals quality, which also deteriorate for high thresholds (the true positives are the proposals predicted correctly and have overlap score over a threshold, so bad localization induce low AUC value).

Since we can not compare results on UCF 101, we only show our results. Figure 12 shows the classification results for UCF 101. On the left we see the mean average precision for different overlap thresholds. On the right we see the average precision per class (we follow [3] evaluation criteria and use 1/8 threshold). We see a big variation in classification results between the classes (*SoccerJugling* and *IcaDancing* with average precision of 1, versus *Skijet* and *Surfing* with average precision of 0.

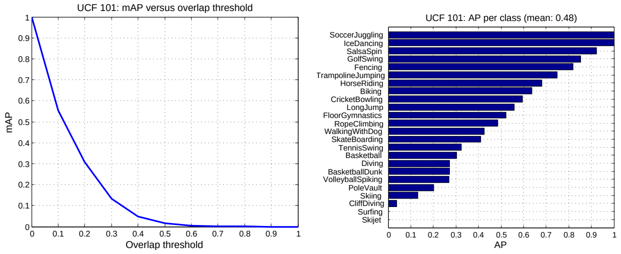


Figure 12. UCF 101 dataset, mean average precision per overlap threshold (left) and average precision per class, for 1/8 threshold (right). We can see big variation between the classes.

Method	Boxing	Clapping	Hand waving	Average
Cao et al.	17.48	13.16	26.71	19.12
SDPM	38.86	23.91	44.70	35.82
Tubelets	**46.00**	**31.41**	85.79	54.40
BING3D	42.86	29.77	**94.73**	**55.79**

Table 3. Average precisions for MSR-II

For MSR-II we have AP scores from three other methods, Tubelets [10], SPDM [26] and Cao et al. [3]. Table 3 shows AP for each of the three MSR-II classes, as well as their average. For the *Boxing* and *Clapping* classes, we perform slightly lower than Tubelets, the former best scoring method, while for the *Hand waving* class, we outperform it by 9%, so on average (mAP score) we still outperform by a bit over 1% over the previous best method.

5. Conclusions

We proposed a new method for spatio-temporal proposals as used for action localization. Our method is called BING3D, as it is a 3D extension of the state-of-the-art BING algorithm for object proposals in still images. The main advantage of BING3D is its fast speed, two orders of magnitude faster than competing methods, that is enabled due to use of simple and fast to compute video features, and a binarization method, that allows the use of quick bitwise operations, in the proposal generation.

We tested BING3D on three benchmark action datasets, and achieved results that are on par or above state-of-the-art on both localization and classification. We presented a thorough evaluation of the method parameters, as well as quantitative and qualitative analysis. We experimented with cross dataset model transfer, where we train our model on one dataset and test it on another, and the results showed that trimmed videos respond differently than untrimmed videos, but within the groups (trimmed/untrimmed) model transfer yields results on par with the model trained on the tested dataset. Thus, training one good model can be sufficient for proposal generation for different datasets.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 2012. 1, 2

[2] M. V. D. Bergh, G. Roig, X. Boix, S. Manen, and L. V. Gool. Online video seeds for temporal window objectness. In *ICCV*, 2013. 2

[3] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1998–2005. IEEE, 2010. 7

[4] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. 1, 2, 3

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 2

[6] I. Everts, J. C. van Gemert, and T. Gevers. Evaluation of color spatio-temporal interest points for human action recognition. *TIP*, 23(4):1569–1580, 2014. 2

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 4

[8] S. Hare, A. Saffari, and P. H. Torr. Efficient online structured output learning for keypoint-based object tracking. In *CVPR*, 2012. 3

[9] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013. 2

[10] M. Jain, J. C. van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014. 1, 2, 5, 6, 7

[11] Z. Jiang, Z. Lin, and L. S. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *TPAMI*, 34(3):533–547, 2012. 5

[12] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2011. 1, 2

[13] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 2

[14] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. 2

[15] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[16] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2003–2010. IEEE, 2011. 4, 7

[17] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003. 2

[18] I. Laptev, M. Marzalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[19] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013. 1, 2

[20] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014. 1, 2, 6

[21] D. Oneata, J. Verbeek, and C. Schmid. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In *ICCV*, 2013. 2

[22] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014. 2

[23] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010. 7

[24] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 1, 2

[25] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *CVPR*, 2013. 2

[26] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 1, 2, 7

[27] D. Tran and J. Yuan. Max-margin structured output regression for spatio-temporal action localization. In *NIPS*, 2012. 2

[28] D. Tran, J. Yuan, and D. Forsyth. Video event detection: From subvolume localization to spatio-temporal path search. *TPAMI*, 2013. 2

[29] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 1, 2

[30] J. C. van Gemert, M. Jain, E. Gati, and C. G. M. Snoek. APT: Action localization Proposals from dense Trajectories. In *BMVC*, 2015. 1

[31] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 2

[32] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013. 4

[33] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *ICCV*, 2013. 2, 4

[34] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008. 2

[35] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012. 1

[36] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009. 2

[37] Q. Zhao, Z. Liu, and B. Yin. Cracking bing and beyond. In *BMVC*, 2014. 5

[38] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2