VIDEO SEGMENTATION VIA BOUNDARY-AWARE FLOW

Ding-Jie Chen, Hwann-Tzong Chen, and Long-Wen Chang

Department of Computer Science, National Tsing Hua University, Taiwan

ABSTRACT

We present a new algorithm for unsupervised video segmentation based on boundary-aware optical flow. Existing video segmentation methods usually tweak their segmentation model to tolerate the inaccuracy in the estimation of optical flow around object boundaries. In contrast, we directly manipulate the optical flow for better quality. We smooth the optical flow via transductive inference to make the flow consistent within the object and fit to the object boundaries. We then use the boundary-aware optical flow to estimate the initial foreground object region from each frame for learning the appearance model. The learned appearance model is consequently used to refine the segmentation result. Experiments on the DAVIS dataset show that our method performs favorably against the existing ones.

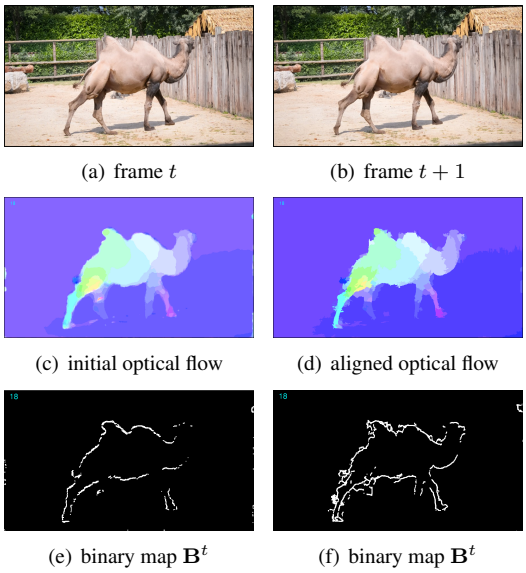
***Index Terms*—** Video segmentation, optical flow, transductive inference

1. INTRODUCTION

Video object segmentation is a fundamental vision problem aiming to distinguish the foreground objects from the backgrounds of video sequences. It is a key component for numerous applications, including video editing, color grading, scene understanding, video summarization, and action recognition [1, 2, 3].

Various algorithms have been presented to segment videos via tracking [4, 5], clustering [6, 7, 8], ranking [9, 10, 11], or propagating [12, 13, 14] at pixel level, superpixel level, or object level. To address the video segmentation task, temporal information should be taken into account to maintain the consistency over the whole video sequence. In practice, the most widely adopted technique for this purpose is optical flow estimation, which models the motion of pixels and can be used to propagate segmentation information among video frames.

We propose a flow smoothing approach and apply it to the unsupervised flow-driven video object segmentation task. The task treats an image region that has different motion from its surrounding regions as a target foreground object. The motion of the foreground object is allowed to be non-homogeneous and thus can be used to segment non-rigid or articulated objects. This task greatly depends on the quality of optical flow for estimating the foreground object. However, the calculation of optical flow is often deteriorated by large displacements or occlusions [15]. Existing video segmentation methods usually tweak their segmentation model to tolerate the inaccuracy of optical flow around object boundaries. In contrast, we directly manipulate the optical flow for the subsequent usage. We borrow the idea from transductive inference and design a flow smoothing method to make the optical flow more suitable for video segmentation. The flow is enforced to be consistent within the object and fit to the object boundaries. Based on the boundary-aware optical flow, we are able to ensure the quality of the initial estimation of potential foreground object regions. We then build a spatio-temporal graphical model of the entire video, and extract all initial foreground object regions from each frame to learn the appearance Gaussian mixture model (GMM). The learned GMM is thus used to gradually refine the segmentations.



(a) frame t (b) frame $t+1$ (c) initial optical flow (d) aligned optical flow (e) binary map $\mathbf{B}^t$ (f) binary map $\mathbf{B}^t$

Fig. 1. An example of smoothed optical flow. (a-b) Two consecutive frames of the sequence `camel`. (c) Optical flow computed using [7] from frame t to $t+1$. (d) The boundary-aware optical flow using the proposed method. The flow at the object boundary is more precise compared with the initial flow. (e) The calculated binary map $\mathbf{B}^t$ using (c). (f) The calculated binary map $\mathbf{B}^t$ using (d).

2. RELATED WORK

The literature on video segmentation can be roughly divided into two categories: semi-supervised methods and unsupervised methods.

2.1. Semi-supervised Methods

Semi-supervised video segmentation methods [4, 5, 6, 12, 13, 14] require the user annotations in some frames and then generate the corresponding segmentations on all other frames. Given some known superpixels, Yang *et al.* [4] and Wen *et al.* [5] track object segments to separate the foreground regions from the background. Grundmann *et al.* [6] oversegment a video sequence into a set of supervoxels and then combine it with the user annotations to select the foreground regions. Some video object segmentation methods require the user to manually annotate a few frames with object segmentations [12, 13, 14] and then propagate these segmentations to all other frames.

2.2. Unsupervised Methods

Unsupervised video segmentation methods [7, 8, 9, 10, 11, 16, 17] exploit the fact that different objects usually have different motions or appearances. Papazoglou and Ferrari [16] propose to segment objects that move differently than their surrounding regions. Brutzer *et al.* [17] assume that the background change slowly and hence consider the pixels that change rapidly to be the foreground. Based on the clustering concept, [7, 8] track keypoints to form trajectories over several frames and then cluster the trajectories for separating the figure-ground keypoints. The methods of [9, 10, 11] rank some combinations from object-like image regions as the potential object segmentations based on the set of object proposals [18, 19].

In sum, the semi-supervised video segmentation methods have better segmentation accuracy but the unsupervised methods enable the processing of large amounts of video sequences without human intervention.

3. APPROACH

The goal of our video segmentation approach is to segment objects with different motion from their surroundings. Our approach includes two main phases, namely *initial foreground estimation* and *segmentation refinement*. The first phase follows the clue from the boundary-aware optical flow and yields an initial foreground region of each frame. The second phase collects all initial foreground regions from each frame to construct the global appearance Gaussian mixture model for segmentation refinement.

3.1. Initial Foreground Estimation

This phase estimates an initial foreground region based on the motion cue. We first compute the optical flow of each pair of consecutive frames, and then the flows are smoothed frame by frame with respect to the global similarity derived from transductive inference over superpixels. Finally, the foreground hypothesis is estimated according to the flow boundaries.

3.1.1. Spatial Graph Construction

We represent each frame t as a set of superpixels $\mathcal{S}^t = \{s_1^t, s_2^t, ..., s_{|\mathcal{S}^t|}^t\}$ using the SLIC algorithm [20]. For each frame t, we define the corresponding weighted connected graph $\mathcal{G}^t = (\mathcal{S}^t, \mathcal{E}^t, \omega)$ with the vertex set $\mathcal{S}^t$ and the edge set $\mathcal{E}^t$. Each edge $e_{ij}^t \in \mathcal{E}^t$ denotes the adjacency relationship between superpixels $s_i^t \in \mathcal{S}^t$ and $s_j^t \in \mathcal{S}^t$. The weighting function $\omega : \mathcal{E} \to [0, 1]$ is defined as

$$\omega_{ij} = e^{-\theta_1 \|c_i - c_j\|_2}, \quad (1)$$

where c_i and c_j denote the RGB mean color of two adjacent superpixels. We can thus define the weight matrix per frame as $\mathbf{W}^t = [\omega_{ij}]_{|\mathcal{S}^t| \times |\mathcal{S}^t|}$. We set parameter $\theta_1 = 60$ for all the experiments.

3.1.2. Optical Flow

We use the flow estimation algorithm in [7] to compute the optical flow between any two consecutive frames t and $t + 1$. For computational efficiency, the GPU implementation [21] can be considered.

3.1.3. Boundary-Aware Flow Smoothing

In video segmentation problem, the optical flow is expected to be smooth within an object and distinct across object boundaries. However, the reliability of flow estimation is often degraded by large displacements or occlusions, particularly around object boundaries. Here, we propose the following scheme to make the optical flow smoother within an object and more fit to the object boundary.

We smooth the optical flow via propagating the flow velocities (with velocity components v_x and v_y) of each superpixel to all other superpixels. For each superpixel, we use the averaged velocity components to represent its superpixel-level optical flow. We then propagate the per-superpixel flow according to the feature similarity between every superpixel pair. In some sense, we would like to smooth the flow velocity more if the two superpixels have higher feature similarity.

The first step of boundary-aware flow smoothing is to calculate the pairwise similarity matrix $\mathbf{A}^t$ for each superpixel set $\mathcal{S}^t$. We apply transductive inference [22] to construct the matrix $\mathbf{A}^t$ from the weight matrix $\mathbf{W}^t$. The similarity matrix $\mathbf{A}^t$ can be defined by

$$\mathbf{A}^t = (\mathbf{D}^t - \theta_2 \mathbf{W}^t)^{-1} \mathbf{I}^t, \quad (2)$$

where $\mathbf{D}^t$ is a diagonal matrix with each diagonal entry equal to the row sum of $\mathbf{W}^t$, θ_2 is a parameter in $(0, 1]$, and $\mathbf{I}^t$ is the $|\mathcal{S}^t|$-by-$|\mathcal{S}^t|$ identity matrix. We set parameter $\theta_2 = 0.99$ for all the experiments. Note that, the weight matrix $\mathbf{W}^t$ describes the feature similarity only between any two *adjacent* superpixels of $\mathcal{G}^t$, but the matrix $\mathbf{A}^t$ globally describes the feature similarity between any two superpixels even if they are not adjacent. Intuitively, propagating per-superpixel flow velocities via $\mathbf{A}^t$ allows us to deal with pairs of uperpixels that are far from each other. Hence, propagating per-superpixel flow velocities via $\mathbf{A}^t$ should provide better smoothness results than propagating via $\mathbf{W}^t$.

Next, the smoothed optical flow $\hat{f}_i^t$ of the superpixel s_i^t in frame t is thus defined as

$$\hat{f}_i^t = \mathbf{D}_{\mathbf{A}^t}^{-1}\mathbf{A}^t \cdot [f_1^t, f_2^t, \ldots, f_{|\mathcal{S}^t|}^t]^T , \tag{3}$$

where $\mathbf{D}_{\mathbf{A}^t}$ is a diagonal matrix with each diagonal entry equal to the row sum of $\mathbf{A}^t$, and $\mathbf{D}_{\mathbf{A}^t}^{-1}\mathbf{A}^t$ means the row normalized version of $\mathbf{A}^t$. Eq. (3) means that the *boundary-aware smoothed flow* of each superpixel is derived from not only its neighboring superpixels but also all other superpixels. Fig. 1 shows one example result of the smoothed flow.

3.1.4. Foreground Hypothesis

Since optical flow usually changes abruptly around the object boundaries, thresholding the gradient magnitude on the flow field can roughly sketch the object region. Given the boundary-aware smoothed flow field $\hat{\mathbf{F}}^t$ of the frame t, we define a binary map $\mathbf{B}^t$ indicating the potential object boundaries as

$$\mathbf{B}_{pq}^t = \begin{cases} 1, & \text{if } e^{-\theta_3 \|\nabla \hat{\mathbf{F}}_{pq}^t\|} < 0.5 \ , \\ 0, & \text{otherwise} \, , \end{cases} \tag{4}$$

where $\nabla \hat{\mathbf{F}}_{pq}^t$ denotes the gradient of smoothed optical flow at row p and column q of the frame t, and hence the binary map $\mathbf{B}^t$ has roughly the same size as the frame t. We set parameter $\theta_3 = 0.7$ in this work. Closed contours in $\mathbf{B}^t$ represent good candidates of foreground regions because a closed contour in $\mathbf{B}^t$ implies having different motion to its surroundings. However, the simple strategy of thresholding in Eq. (4) often results in incomplete boundaries. Inspired by the point-in-polygon problem in computational geometry [23], Papazoglou and Ferrari propose an efficient *integral intersections* algorithm [16] that can identify the pixels inside incomplete boundaries. We use the integral intersections algorithm to indicate the pixels inside the boundaries of $\mathbf{B}^t$. After applying integral intersections, a superpixel with high proportion of indicated pixels means the superpixel has high potential to be the foreground. Consequently, the high-foreground-potential superpixels should have larger impacts when learning the appearance model. Fig. 1 shows one example result of the binary map $\mathbf{B}^t$.

3.2. Segmentation Refinement

We collect all high-foreground-potential superpixels from each frame to construct a Gaussian mixture model of global appearance. In contrast, superpixels with very low foreground potentials can be used to construct the background appearance model. Further, we define an energy function to encourage the spatio-temporal smoothness for refining the segmentation over the entire video sequence.

3.2.1. Spatio-Temporal Graph Construction

In Section 3.1.1, we have defined the spatial graph $\mathcal{G}^t = (\mathcal{S}^t, \mathcal{E}^t, \omega)$ for each frame t. Likewise, in the *spatio-temporal* graph $\mathcal{G}$, two superpixels $s_i^t \in \mathcal{S}^t$ and $s_j^{t+1} \in \mathcal{S}^{t+1}$ are connected if s_i^t can cover s_j^{t+1} via the smoothed optical flow.

3.2.2. Energy Function

We may formulate the procedure of video segmentation as a binary labeling problem with the foreground and background labels. Each superpixel s_i^t can be assigned with one label $l_i^t \in \{0, 1\}$. The labeling $\mathcal{L} = \{l_i^t\}$ among all superpixels in the given video can be evaluated by some energy function [9, 16, 24]. Here, we define the energy function as

$$E(\mathcal{L}) = E_A + \alpha_1 E_L + \alpha_2 E_S + \alpha_3 E_T \, . \tag{5}$$

The data term E_A in Eq. (5) evaluates how likely a superpixel belongs to the foreground GMM or background GMM. The second data term E_L in Eq. (5) is used to encourage foreground labeling in areas where independent motion has been observed. The spatial smoothness term E_S and the temporal smoothness term E_T encourage spatial and temporal smoothness, respectively. The parameters $\alpha_1, \alpha_2, \alpha_3$ are the weights for different terms. The energy function can be optimized using the graph-cut algorithm. For more details about the design of energy function and the optimization, please refer to [16].

Notice that, the superpixel-level segmentation may not well align with the object boundary. After obtaining the segmentation, we further use the guided filter [25] to reduce the under-segmentation error derived from oversegmentation.

4. EXPERIMENTAL RESULTS

We compare our approach with several state-of-the-art unsupervised video segmentation methods: NLC [27], CVOS [28], TRC [29], MSG [7], KEY [9], SAL [30], and FST [16]. The evaluations are performed with respect to the three metrics suggested in the dataset DAVIS [26], namely, region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$), and temporal stability ($\mathcal{T}$). In our experiments, all parameters are fixed without further tuning. We use roughly $2{,}000$ superpixels per frame.

We evaluate our approach on the dataset DAVIS [26]. It contains 50 high-resolution sequences covering a wide range

Table 1. Quantitative comparison (%) of region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$), and temporal instability ($\mathcal{T}$) on DAVIS [26]. The 'mean' is the average dataset error. The 'recall' measures the fraction of sequences scoring higher than a threshold. The 'decay' quantifies the performance loss (or gain) over time. For rows with an upward pointing arrow, the higher numbers are better, and vice versa for rows with a downward point arrow. The best two scores of each dataset are colored in red and green.

Metrics	NLC [27]	CVOS [28]	TRC [29]	MSG [7]	KEY [9]	SAL [30]	FST [16]	Ours
mean $\mathcal{J}\uparrow$	64.1	51.4	50.1	54.3	56.9	42.6	57.5	**62.5**
recall $\mathcal{J}\uparrow$	73.1	58.1	56.0	63.6	67.1	38.6	65.2	**73.6**
decay $\mathcal{J}\downarrow$	8.6	12.7	5.0	2.8	7.5	8.4	4.4	**-0.5**
mean $\mathcal{F}\uparrow$	59.3	49.0	47.8	52.5	50.3	38.3	53.6	**57.5**
recall $\mathcal{F}\uparrow$	65.8	57.8	51.9	61.3	53.4	26.4	57.9	**65.0**
decay $\mathcal{F}\downarrow$	8.6	13.8	6.6	5.7	7.9	7.2	6.5	**2.9**
mean $\mathcal{T}\downarrow$	35.6	24.3	32.7	25.0	19.0	60.0	27.6	**20.7**

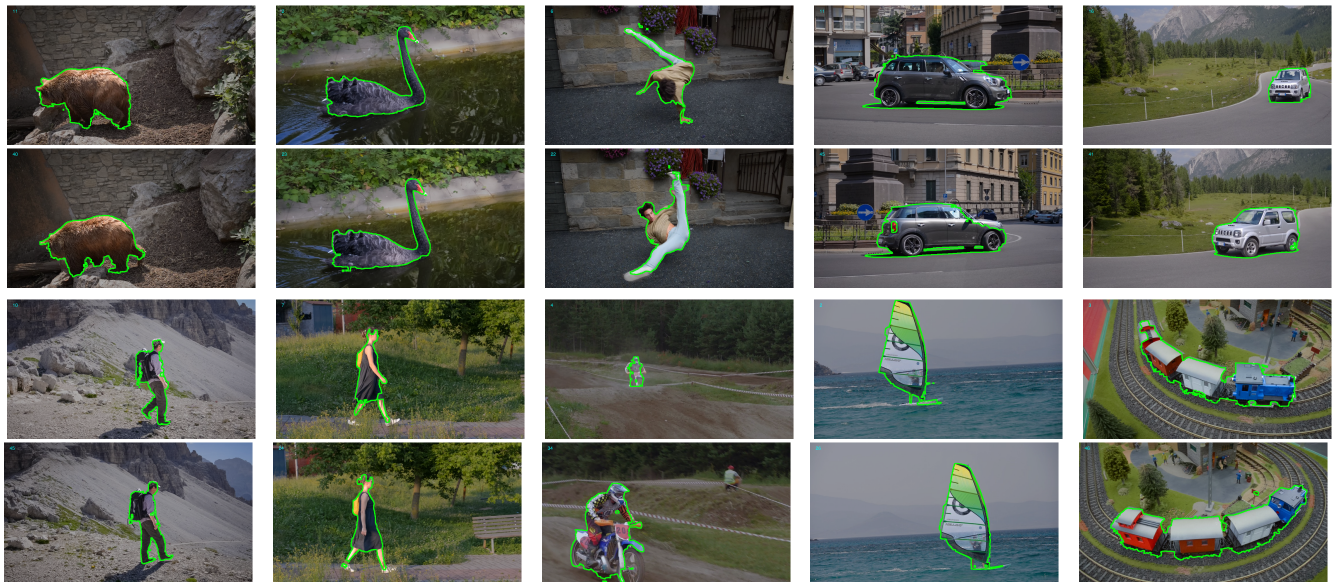


Fig. 2. Qualitative video segmentation results from some sequences of DAVIS [26]. Our method demonstrates robustness to some challenging scenarios such as complex objects and fast-motion.

of object segmentation challenges. Table. 1 summarizes the average performance over the entire dataset. As can be seen in Table. 1, our method outperforms all other unsupervised video segmentation methods excepts NLC in some entries. Our method achieves the best performance on the recall $\mathcal{J}$, decay $\mathcal{J}$, and decay $\mathcal{F}$, and performs comparably on the other measures. Since the 'decay' quantifies the performance loss (or gain) over time, the good performance on this measure demonstrates that our method usually has better consistent segmentation performance than all other methods among the entire video sequence. Fig. 2 shows the qualitative results of video segmentation.

5. CONCLUSION

We have shown that the boundary-aware flow smoothing method can generate useful optical flow specifically for the video segmentation task. With the aids of the improved optical flow result, the video segmentation task can extract high-foreground-potential superpixels for learning the GMM appearance model, which is helpful in segmentation refinement. The experimental results also show that the proposed video segmentation method, which benefits from the boundary-aware flow, performs favorably against the existing methods.

6. REFERENCES

[1] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri, "Actions as space-time shapes," *IEEE TPAMI*.

[2] Aastha Jain, Shuanak Chatterjee, and René Vidal, "Coarse-to-fine semantic video segmentation using supervoxel trees," in *ICCV*, 2013.

[3] Anna Khoreva, Fabio Galasso, Matthias Hein, and Bernt Schiele, "Classifier based graph construction for video segmentation," in *CVPR*, 2015.

[4] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang, "Robust superpixel tracking," *IEEE TIP*, 2014.

[5] Longyin Wen, Dawei Du, Zhen Lei, Stan Z. Li, and Ming-Hsuan Yang, "JOTS: joint online tracking and segmentation," in *CVPR*, 2015.

[6] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan A. Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR*, 2010.

[7] Thomas Brox and Jitendra Malik, "Object segmentation by long term analysis of point trajectories," in *ECCV*, 2010.

[8] Peter Ochs and Thomas Brox, "Higher order motion models and spectral clustering," in *CVPR*, 2012.

[9] Yong Jae Lee, Jaechul Kim, and Kristen Grauman, "Key-segments for video object segmentation," in *ICCV*, 2011.

[10] Tianyang Ma and Longin Jan Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *CVPR*, 2012.

[11] Dong Zhang, Omar Javed, and Mubarak Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *CVPR*, 2013.

[12] Brian L. Price, Bryan S. Morse, and Scott Cohen, "Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *ICCV*, 2009.

[13] Tinghuai Wang and John P. Collomosse, "Probabilistic motion diffusion of labeling priors for coherent video segmentation," *IEEE TMM*, 2012.

[14] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black, "Video segmentation via object flow," in *CVPR*, 2016.

[15] Thomas Brox and Jitendra Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE TPAMI*, 2011.

[16] Anestis Papazoglou and Vittorio Ferrari, "Fast object segmentation in unconstrained video," in *ICCV*, 2013.

[17] Sebastian Brutzer, Benjamin Höferlin, and Gunther Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *CVPR*, 2011.

[18] Ian Endres and Derek Hoiem, "Category independent object proposals," in *ECCV*, 2010.

[19] João Carreira and Cristian Sminchisescu, "CPMC: automatic object segmentation using constrained parametric min-cuts," *IEEE TPAMI*, 2012.

[20] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE TPAMI*, 2012.

[21] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *ECCV*, 2010.

[22] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[23] James D. Foley, Andries van Dam, Steven Feiner, and John F. Hughes, *Computer graphics - principles and practice, 2nd Edition*, Addison-Wesley, 1990.

[24] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM TOG*, 2004.

[25] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *IEEE TPAMI*, 2013.

[26] Federico Perazzi, Jordi Pont-Tuset, B. McWilliams, Luc J. Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *CVPR*, 2016.

[27] Alon Faktor and Michal Irani, "Video segmentation by non-local consensus voting," in *BMVC*, 2014.

[28] Brian Taylor, Vasiliy Karasev, and Stefano Soatto, "Causal video object segmentation from persistence of occlusions," in *CVPR*, 2015.

[29] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi, "Video segmentation by tracing discontinuities in a trajectory embedding," in *CVPR*, 2012.

[30] Wenguan Wang, Jianbing Shen, and Fatih Porikli, "Saliency-aware geodesic video object segmentation," in *CVPR*, 2015.