Crowd-Guided Ensembles: How Can We Choreograph Crowd Workers for Video Segmentation?

Alexandre Kaspar
MIT
akaspar@mit.edu

Geneviève Patterson
Microsoft Research NE
gen@microsoft.com

Changil Kim
MIT
changil@csail.mit.edu

Yağız Aksoy
ETH Zurich
yagiz@mit.edu

Wojciech Matusik
MIT
wojciech@mit.edu

Mohammad Elgharib
Hamad Bin Khalifa University
melgharib@hbku.edu.qa

Figure 1. An illustration of our two proposed crowd-guided ensemble methods. Left: Our segmentation ensemble combines the results of multiple crowd workers through the guidance of an oracle reviewer. Right: Our propagation ensemble gathers the information about where multiple distinct algorithms fail from the accumulated scribbles of crowd workers, and merges it into the result that incorporates the best of each algorithm.

ABSTRACT
In this work, we propose two ensemble methods leveraging a crowd workforce to improve video annotation, with a focus on video object segmentation. Their shared principle is that while individual candidate results may likely be insufficient, they often complement each other so that they can be combined into something better than any of the individual results—the very spirit of collaborative working. For one, we extend a standard polygon-drawing interface to allow workers to annotate negative space, and combine the work of multiple workers instead of relying on a single best one as commonly done in crowdsourced image segmentation. For the other, we present a method to combine multiple automatic propagation algorithms with the help of the crowd. Such combination requires an understanding of where the algorithms fail, which we gather using a novel coarse scribble video annotation task. We evaluate our ensemble methods, discuss our design choices for them, and make our web-based crowdsourcing tools and results publicly available.

INTRODUCTION
Video segmentation is one of the most essential tools for movie post-production and more recently for generating training data for a multitude of data-driven algorithms. The current practice heavily depends on specialized rotoscoping artists who utilize several commercial software products, often in orchestration. The dependence on specialized artists results in an excessive financial cost especially for generating segmentations for many videos due to the highly time-consuming nature of rotoscoping. In this paper, we aim to democratize rotoscoping by simplifying the work of the artist into a less intensive, reviewing role while decreasing the overall cost of the video segmentation task. We achieve this with a distributed workflow consisting of two novel ensemble methods that leverage multiple annotations from inexperienced crowd workers.

Crowdsourcing is a widely used tool for distributing large manual tasks to a group of inexperienced workers. In visual data processing, it is widely used for 2D image-space operations like image segmentation. In the previous efforts for crowdsourced image segmentation [4, 5, 24, 34], the segmentation results are accepted from a single worker’s result in its entirety, discarding the efforts of the rest. This approach wastes the full potential of the crowd since it only keeps the result of a single best worker. Moreover, these methods do not trivially extend to video segmentation as it requires a careful treatment of temporal coherency, and a frame-by-frame application on multiple frames may result in a prohibitive cost and thus not...
be scalable. The endeavor to propagate image segmentation to videos is not mature enough and a general solution to this problem is under active research [30].

We propose a novel crowdsourced video segmentation workflow that allows and deliberately utilizes the redundancy of input, whether it is from crowd workers or from automated propagation methods. We make use of two novel ensemble methods in two key steps frequently arising in modern video segmentation pipelines [1, 8], namely keyframe segmentation and propagation (see Figure 1). These directly focus on two important aspects of video segmentation: (1) increasing the segmentation quality of the final single-frame segmentation by merging results from a group of unskilled crowd workers; and (2) addressing the issue of scalability, i.e. how to use the temporal dependencies of videos to decrease the average effort needed for each frame. We further provide improvements on the segmentation interface that is presented to the crowd workers. Our contributions consist of:

- The acquisition of higher-quality segmentation by merging the work of multiple crowd workers to produce a result which is better than any of the individual results,
- The improvement of crowdsourced segmentation capabilities with the introduction of negative space annotation and the validation of its positive adoption by crowd workers,
- A novel coarse scribbling task results of which we use to merge multiple automated propagations locally using the guidance of the crowd,
- The application of our scribble annotations to automating segmentation review, and further to propagating trimaps for video matting, illustrating the flexibility of our methods.

We evaluate our methods and design decisions thoroughly and discuss the merits and shortcomings of our methods. We make our tools, web-based user interfaces, and annotation results public to facilitate further research; see anonymized url for submission.

RELATED WORK

A canonical system for human-in-the-loop video object segmentation is rotoscoping [8]. An input video is isolated to cuts, which are then decomposed into a set of keyframes. The keyframes are manually segmented first, these segmentations are propagated to in-between frames automatically using motion cues and image features, and then the propagated segmentations are refined manually [1]; this process is repeated until the desired quality is achieved. Since rotoscoping is done by highly skilled artists and the pace of the process is rather slow—fifteen frames per day on average by an experienced artist [23]—this annotation pipeline is cost-intensive and seldom scalable. Nonetheless, rotoscoping is a crucial tool for creating high-quality object segmentation, which is regularly practiced in diverse areas from simple image composition to visual effects used for movie productions to the generation of ground truth annotation on which today’s artificial intelligence (AI) engines depend. It is therefore of great importance to make the process both more scalable and more accessible.

Crowdsourced Image Segmentation

Recent crowdsourcing systems for image segmentation include OpenSurfaces [5], upon which our segmentation interface is based; Intrinsic Images in the Wild [4]; Materials in Context [6]; the hierarchical instance segmentation pipeline of Microsoft COCO [24]; and the Video Annotation Tool from Irvine, California [34]. They share a common strategy: validating a single worker’s annotations. Our approach embraces the small, local errors that different annotators or automatic methods make and relies on workers to combine multiple segmentations into a higher-quality result. Many crowd-annotation systems improve on challenges such as scalability and cost-effectiveness [12, 27, 33], worker consensus [15, 14], annotation quality [16], crowd-AI interaction algorithms [7, 28, 22], identifying breakage in automated labeling [36], and workflow control [11]. In their paper on the future challenges facing crowd work, Kittur et al. [21] emphasize that crowd-guided AI systems are a core problem of this field. This paper presents the first attempt to have crowd workers guide automatic video segmentation.

Video Segmentation

Segmentation propagation plays a crucial role for reducing the workload of per-frame segmentations, and is extensively studied in the video segmentation literature [10, 29, 25, 31]. While the most recent approaches are increasingly using more complicated data-driven models, they are also relying on the availability of video segmentation datasets. The recent benchmark of Perazzi et al. [30] points out the scarcity of video segmentation datasets as well as their limited size and variations. In this work, we do not aim to compete with the automatic propagation methods; rather, our goal is to facilitate the acquisition of the annotated data which these methods require, by providing a more scalable way to create larger-scale annotated datasets. Moreover, the diversity of these automated propagation techniques is what our ensemble method builds on. Our results show that even with a set of simple propagation methods with clear shortcomings, the crowd workforce can help generate propagation results that often overcome the limitations of each individual method.

ENSEMBLE METHODS USING THE CROWD

Ensemble methods combine multiple results to achieve higher quality than any of the individual results [13]. In crowdsourcing systems, it is common practice to ensemble the results of tasks which exhibit high variance inherent to the varying degrees of human skills, attentions or complex internal motivations. The most common approach is to make a Bayesian decision, which defaults to a simple voting scheme with votes possibly weighted by their confidence if such information is available.

We propose the use of novel ensemble strategies in two components of our video segmentation pipeline: first for individual frame segmentations and second in merging multiple corrections from the crowd in propagated segmentation results. We do so with the considerations to generate high quality video segmentation results from the crowd.
We developed a polygon-based segmentation interface based on the idea that workers have the option to use negative polygons that subtract regions from the foreground in addition to the standard, positive polygons. The idea is motivated by the extensive use of negative space in rotoscoping [8]. In practice, this allows higher-quality segmentations as real, complex objects often contain small regions where the background is visible as illustrated in Figure 2. Defining complex polygons with negative annotations introduces a process ambiguity: one can segment only the positive space by decomposing it into multiple components, or one can segment the whole target as one positive region that possibly includes several negative components. We solve this ambiguity in our merging stage and do not dictate which strategy to use as both have advantages and disadvantages.

Figure 2. Our segmentation user interface with positive and negative polygons for the boat sequence (top-left) and the corresponding segmentation overlaid (bottom-right).

**Segmentation Ensembles**

The standard approach for single-frame segmentation acquisition in large crowdsourcing efforts such as OpenSurfaces [5] and COCO [24] consists of using a secondary task to evaluate the quality of each individual segmentation. If any single segmentation is evaluated as sufficient, it is accepted, else it is rejected and a new segmentation is requested. With a known set of expert segmenters [26] or qualified workers [5], the evaluation task can be sidestepped and the segmentation is directly accepted. While this approach results in a small cost per segmentation, it may not be sufficient for high-quality results using the crowd because no single worker may ever reach the desired quality. Furthermore, the evaluation strategy assumes some quality threshold decided by the crowd which is highly subjective as noted in OpenSurfaces [5] and may not necessarily match the desired quality.

Instead of using pass/fail evaluations of individual segmentations, we propose a system that is designed to take advantage of multiple distinct segmentations to achieve a high quality final result. Furthermore, we introduce the use of negative annotations, which makes it easier for the segmenters to generate detailed results such as the one shown in Figure 2.

**Negative Space Annotation**

We developed a polygon-based segmentation interface based on OpenSurfaces [5] with one major modification: the crowd workers have the option to use negative polygons that subtract regions from the foreground in addition to the standard, positive polygons. The idea is motivated by the extensive use of negative space in rotoscoping [8]. In practice, this allows higher-quality segmentations as real, complex objects often contain small regions where the background is visible as illustrated in Figure 2. Defining complex polygons with negative annotations introduces a process ambiguity: one can segment only the positive space by decomposing it into multiple components, or one can segment the whole target as one positive region that possibly includes several negative components. We solve this ambiguity in our merging stage and do not dictate which strategy to use as both have advantages and disadvantages.

**Merging Segmentations**

Given $N$ segmentations of the same object by distinct segmenters, we obtain our segmentation result in our review phase. We assume the existence of an oracle, who can be a single user or the crowd, who provides us with weights $w_i$ that represent the relative qualities of each segmentation $i = 1, \ldots, N$.

When dealing with complex segmentations made of multiple polygons, treating the whole segmentation as a single merging operation breaks polygons that have no counterpart in the other segmentations. Thus, we first cluster polygons into minimal disjoint clusters such that no polygon intersects a polygon from another cluster. Furthermore, we treat each polygon class (positive or negative) as a separate layer (foreground or background) and merge polygons within their cluster and class separately. This is especially important to make full use of the negative polygons, which our experiments showed to be an indication of higher quality. Given a specific class layer and cluster $C$, we take per-worker subsegmentations made of their corresponding polygons, and merge them pixel-wise by weighted average voting:

$$M = \left[ \frac{1}{W} \sum_{i=1}^{N} w_i M_i \right]_{0.5},$$

where $[\cdot]$ denotes a thresholding operator resulting in a binary value in $\{0,1\}$, $M_i \in C$ a subsegmentation of worker $i$, and $W = \sum_i w_i$ the total weight. Finally, we take the union of the non-overlapping clusters that were generated by a weighted majority of workers, and merge the foreground layer $F$ with the background layer $H$ into the full segmentation $F \cap \neg H$.

While our strategy could be suboptimal in isolated cases, it minimizes the work of the reviewer and already achieves a significant overall quality improvement over using only one of the input segmentations, or using a simple voting strategy. An example merging result can be seen in Figure 3.

**Oracle Review**

In our implementation, the oracle selecting the individual weights $w_i$ is the requester (or reviewer) who accepts or rejects results, and distributes the money to the crowd workers. The review process reduces to choosing weights for each of the $N$
results. Selecting the weights \(w_i = 1 + \beta_i\) serves two purposes: (i) to select the best result composition with weights \(w_i\), and (ii) to supplement the base task reward with additional financial bonuses \(\beta_i\) (in cents).

When crowdsourcing large segmentation acquisitions, an extra problem arises with the reward selection. Online platforms such as Amazon Mechanical Turk require the selection of a reward to publish a task. This requirement implies that the requester must evaluate the complexity of the task to choose an appropriate reward. Instead of relying on some complexity assessment, we use a low base reward, which is then complemented with financial bonuses \(\beta_i\) directly derived from the merging weights chosen by the oracle during review. The base reward encapsulates both the maximal amount of money we are giving for any segmentation and the minimum reward that complex segmentations require to attract enough good segmenters.

**Propagation Ensembles**

Given the keyframe segmentations, a standard rotoscoping pipeline attempts to propagate them to the neighboring frames. We assume that we have a number of algorithms available to do this, and that they each make different assumptions so that there are variances among their propagation results.

As it is unlikely for any single segmentation propagation method to produce satisfactory results for all frames in all scenarios, we define a new segmentation merging strategy. The most straightforward way to merge them is by per-pixel majority voting. This, however, does not include any knowledge of where different methods likely fail, and is only effective when the majority of the methods perform well. Instead, our strategy makes use of weak annotations from the crowd that describe where the different methods fail. We observe that these annotations are much easier for the crowd workers than conducting a full per-frame segmentation. Thus, more workers can be assigned to corrections of a frame while still preserving cost-effectiveness especially when compared to conducting a full per-frame segmentation.

**Scribble Annotation**

Our interface for this task includes a slider bar to go over the frames of the video interval and two sets of buttons: one to vary the opacity of the overlaid segmentation result on the input frame, and the other to set the size of the scribble brush as shown in Figure 4.

For each of \(K\) different propagations given for a frame, we first merge the crowd scribbles to create scribble heat maps \(S^k\):

\[
S^k = \frac{1}{B} \sum_{i=1}^{N} b^k_i S^k_i , \tag{2}
\]

where \(b^k_i \in [0.25, 1]\) is a weight inversely proportional to the *average brush size* used by the worker \(i\) for the propagation algorithm \(k\), \(S^k_i\) is the corresponding scribble map, and \(B = \sum_i b^k_i\). The average brush size for the scribble \(S^k_i\) is computed as the average of stroke brush sizes weighted by their corresponding stroke length.

We then merge the scribble heat maps and the propagation results \(P^k\) to get our segmentation result:

\[
M = \left[ \sum_{k=1}^{K} P^k \circ S^k \right]_{\geq A} . \tag{3}
\]

where \(\circ\) is a pixelwise operator and \(A\) is a pixelwise threshold. We investigated three different interpretations of our scribble task, each of which corresponds to a different merging operation \(\circ\). Each of these has pros and cons which we detail in the experiment evaluations.

**Scribble as Error Correction**

We use the scribbles to invert the erroneous regions of segmentation propagations. The merging operator \(\circ\) is defined as

\[
p^k \circ q^k = \begin{cases} 
q^k - q^k & \text{if } p^k = 1 \\
q^k & \text{if } p^k = 0
\end{cases} \tag{pxor}
\]

where \(p^k \in \{0, 1\}\) denotes a pixel of the binary mask \(P^k\), \(q^k \in [0, 1]\) is from the scribble map \(S^k\), and the threshold \(A = K/2\), i.e. the inversion only occurs if the majority requires it.

**Scribble as Soft Penalty**

We use the scribbles to locally penalize a method. This can be interpreted as annotating the regions where we do not trust the segmentation propagation. The merging operator \(\circ\) is here defined as

\[
p^k \circ q^k = p^k (1 - q^k) , \tag{wmaj}
\]

where we use the per-pixel threshold \(A = ([K] / 2)^{1/2}\) and power \(\alpha = 2\) for smoothness.

**Scribble as Segmentation Refinement**

We use the scribbles to locally overwrite the segmentation. In this scenario, we add a brush type selection. Users can use two different brushes to either scribble foreground or background. When a new brush stroke overlaps with an old one, the overlap region is replaced so that every pixel of a worker scribble is either positive (foreground), negative (background) or undefined (no scribble). In this case, merging consists of generating two different scribble heat maps \(S^k_+\) and \(S^k_-\) for the foreground respectively the background, and using the following \(\circ\) merging operator

\[
p^k \circ (q^k_+, q^k_-) = \begin{cases} 
1 & \text{if } q^k_+ = 1 \\
0 & \text{if } q^k_- = 1 \\
p^k & \text{otherwise}
\end{cases} \tag{2-brushes}
\]
We evaluate our two proposed ensemble methods in the context of a crowdsourced pipeline that follows the conventional keyframe segmentation and propagation strategy of rotoscopy. The evaluation is done using the DAVIS dataset [30] and its three metrics: region similarity $J$, contour accuracy $F$, and temporal stability $T$. The first two metrics $J$ and $F$ respectively measure the amount of correct pixelwise overlap of segmentations (commonly referred to as intersection over union), and the quality of the segmentation boundaries. The values are each in the interval of [0, 1]; the higher the better. The last metric $T$ measures the temporal smoothness, for which lower values correspond to better temporal transitions of segmentations. The dataset consists of 50 short video sequences that have been annotated by a rotoscoping artist, for a total of 3455 frames at 1080p HD resolution.

The propagation part of the evaluation is based on our crowdsourced segmentation results. This differs from the typical evaluation of semi-supervised and unsupervised methods on DAVIS in that we cannot assume we have access to the ground truth since our goal is to have the crowd generate it. Thus, for all our automated propagations, the training data we use comes from the results of our crowd workers and not from the original DAVIS dataset.

All of our experiments were done on the Amazon Mechanical Turk platform. For all the evaluation figures including those related to scribbles, we used the $pxor$ merging method with brush regularization $f = 1/2$ unless stated otherwise.

We refer the readers to the supplementary material accompanying our paper for screenshots, videos, and the code demonstrating example sessions using our crowdsourcing user interface, whenever they are mentioned throughout the rest of the paper. Detailed descriptions about the files included and additional analysis are found there as well.

**Segmentation Experiments**

Our baseline segmentation is acquired by sending each segmentation to 3 different workers. We initially allowed workers with a global success rate higher than 50% to work on our segmentation tasks, and refined our worker group later using a whitelisting strategy, where workers were assigned a custom accreditation maintained by us. Our segmentation results have been generated by a set of 70 best-performing workers.

The instructions of this task contain both (1) generic segmentation instructions including examples of good and bad
workers created 34,761 polygons, of which 19,985 (57%) were negative polygons. The number and percentage of negative polygons vary significantly with sequences. For instance, the simple blackswan sequence of DAVIS required 8 negative polygons out of 172 polygons (4%), whereas the more complex boat sequence consisted mostly of negative polygons (4006 of total 4409 polygons, amounting to over 90%; see Figure 2 for an example segmentation). The presence of negative annotations in a segmentation weakly correlates with its J value being above average for that segmentation ($r = 0.17, p < 0.01$) and similarly with its F value ($r = 0.26, p < 0.01$).

**Segmentation Quality**

In Table 1, we provide the results for the naïve approach of fully segmenting the video sequences through keyframe segmentation tasks. Two major observations are: (1) the naïve full segmentation does not ensure any temporal coherence—our workers perform independent segmentations—and thus a high performance in the temporal stability metric T is not expected; and (2) segmentations done by workers do not always yield similar level of segmentation quality around fine details such as thin occluders and intricate boundaries.

---

**Use of Negative Space Annotation**

The concept of negative space is standard in composition in the visual arts. However, to the best of our knowledge, it is the first time to be used in a crowdsourced interface for high-quality image segmentation, which then raises the question: can workers make good use of it?

With all worker segmentations of all sequences counted, the oracle user using the crowd workers created 34,761 polygons, of which 19,985 (57%) were negative polygons. The number and percentage of negative polygons vary significantly with sequences. For instance, the simple blackswan sequence of DAVIS required 8 negative polygons out of 172 polygons (4%), whereas the more complex boat sequence consisted mostly of negative polygons (4006 of total 4409 polygons, amounting to over 90%; see Figure 2 for an example segmentation). The presence of negative annotations in a segmentation weakly correlates with its J value being above average for that segmentation ($r = 0.17, p < 0.01$) and similarly with its F value ($r = 0.26, p < 0.01$).

**Segmentation Quality**

In Table 1, we provide the results for the naïve approach of fully segmenting the video sequences through keyframe segmentation tasks. Two major observations are: (1) the naïve full segmentation does not ensure any temporal coherence—our workers perform independent segmentations—and thus a high performance in the temporal stability metric T is not expected; and (2) segmentations done by workers do not always yield similar level of segmentation quality around fine details such as thin occluders and intricate boundaries.
The table further shows the effect of our merging procedure for the segmentation task when compared to alternative strategies. This justifies our use of multiple results instead of a single one, and also shows that our merging strategy performs better than the two obvious alternatives: using a uniform average, i.e. the same weight for every worker’s result; or using the single best result, i.e. reject all but the best. Note that our method reduces gracefully to the second alternative in the presence of an outstanding worker.

Impact of Replication
Figure 6 shows the evolution of the segmentation quality as we increase the replication count from $R = 1$ to 10. Odd counts tend to be better since they avoid ties during majority voting. Our clustered merging strategy seems to be always better than the default pixelwise merging, and does especially better for even replication counts. However, clustering does not seem to improve quality substantially for odd replication counts. The whitelisting strategy did not improve coverage ($J$), but did improve boundary accuracy ($F$). This suggests that replication helps with coverage, but higher skills are needed for finer details.

Automating the Review
Figure 6 also shows the result of using scribbles for automating the review. It does not seem to produce a significant coverage ($J$) improvement to the default pixelwise averaging, but it does significantly improve the boundary accuracy ($F$). While the crowd does not reach the level of a dedicated oracle user, our results show that it is a viable alternative if the pipeline is desired to be more automated and scalable through crowdsourcing.

Propagation Experiments
In order to evaluate our propagation ensemble method, we make use of two complementary classes of propagation methods. Note that our method is agnostic to which propagation algorithm is used and the choices presented below can easily be replaced with any future propagation algorithm. In our experiments, the keyframe segmentations are propagated to the others using two classes of algorithms detailed below.

Optical flow–based propagation: we warp a given segmentation at frame $t$ to frames $t ± k$ using the optical flow computed between them with large-displacement optical flow [9]. The flow is computed forward and backward, resulting in two distinct propagations.

Feature–based classification: to complement the smooth flow-based approach, we use deep feature classification. We extract hypercolumn features [17] which we compute with VGG16 available in MatConvNet [32] to train a Gaussian-kernel support vector machine (SVM) classifier. We include a simple attention model that consists of applying this method in two stages. During the first stage, the result is used to localize the segmentation target. The segmentation is done using the localization result to focus the classification around the target.

To verify that our scribble task is cost-effective (i.e. it provides advantages to doing full segmentation in terms of quality versus price), we consider the crowdsourced segmentations every $P$ frames as keyframes, from which we propagate the segmentation using each of the aforementioned methods. We then collect ten different scribbling results over each frame using our scribble task. One task assignment consisted of annotating the three candidate propagation intervals ($P$ frames, each for all three propagation candidates). We used uniform keyframe samplings $P = 25, 10, 5, 3$ for an unbiased evaluation. The task reward was $0.01 per single frame scribbling.

To evaluate the impact of the number of propagation methods, we additionally used three recent propagation techniques: Bi-lateral Video Segmentation [25] that propagates the segmentation using the bilateral grid; Video Propagation Networks [20] that uses a neural network with bilateral convolution layers; and the unsupervised technique FusionSeg [18] that uses objectness [19] to propose an object segmentation. Each of the corresponding propagations (for $P = 25, 10, 5, 3$) was acquired similarly to the original experiment, but only over the validation sequences of DAVIS (20 out of 50). Furthermore, instead of annotating the three methods at once, each method was annotated in different tasks. Finally, note that the unsupervised method FusionSeg [18] does not require a frame to propagate from, but we still created tasks with sequences of corresponding length $P$ so as to evaluate the impact of task load.

Propagation Results
Regarding the scribble task, we first verify its positive impact on the propagated segmentation quality and how such increase in quality compares to a denser sampling of frames for individual segmentation with respect to the cost. We then evaluate scribble design components including the use of different brush sizes, the impact of brush regularization, different merging strategies, and the merging order relative to scribble acquisition. Finally, we consider the impact of increasing the number of propagation methods and the task load.

Quality and Cost Tradeoffs
The performance improvement that comes from using the scribbles during merge is demonstrated in Figure 7, where we performed the evaluation on all 50 sequences of DAVIS [30] to report the average error metrics at each sampling interval $P = 25, 10, 5, 3$. The cost of each step is summarized in Table 2. Cost efficiency is achieved since the quality increase from the scribbles is higher than the one from increased sampling. We note that the penalty merging $\text{wmaj}$ is more stable than the corrective merging $\text{pxor}$. The former always increases the quality whereas the later becomes detrimental as the sampling reaches small intervals ($P = 5$ and $P = 3$).

<table>
<thead>
<tr>
<th>Task</th>
<th>Segmentation</th>
<th>Scribble</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame cost</td>
<td>$0.15</td>
<td>$0.01</td>
</tr>
<tr>
<td>Replication</td>
<td>$\times 3$</td>
<td>$\times 10$</td>
</tr>
<tr>
<td>Bonus (on average)</td>
<td>$0.15</td>
<td>-</td>
</tr>
<tr>
<td>Total cost (on average)</td>
<td>$0.90</td>
<td>$0.10</td>
</tr>
</tbody>
</table>

Table 2. Per-frame, per-worker cost breakdown. Note that some tasks consist of multiple frames and we report the effective cost-per-frame here. The bonus is taken as an average.
Impact of Brush Size
We evaluated the impact of the initial brush size and put this in perspective with the distribution of brush sizes that workers used given the quality of the result they contributed to. We provided three different brush sizes corresponding to radii of $b \in \{8, 16, 32\}$ screen pixels and re-ran the original scribble experiment at sampling intervals $P = 25$ with the initial brush size being the smallest this time in contrast to the largest being default. Selecting the smallest brush size initially led to a lower average brush size being used: $E[b] = 22.1$ when the initial brush size was $b_0 = 8$ versus $E[b] = 26.2$ when the initial brush size was $b_0 = 32$. However it also led to a slightly lower quality as summarized in Table 3.

In Figure 9, we show the joint distribution of brush sizes and corresponding result quality. We can observe three peak concentrations of single brush sizes being used for $b = 8, 16$ or 32. Workers who use a mix of different sizes produce better results on average ($J = 0.82, F = 0.85$) than those using a single brush ($J = 0.80, F = 0.83$). However, they only accounted for 2.7% of the total scribble work (2,875 out of 105,641 valid assignments).

Impact of Brush Regularization
Brush regularization seems to have a consistent impact on the quality as shown in Figure 8. The best results are obtained with regularization $f \in [0.5, 0.8]$. Note also that the penalty-based merging strategy is generally more stable and the regularization has less impact on it. However, most of the best results are obtained with the corrective strategy given a low regularization $f < 1$.

Impact of Merging Order
Table 4 shows the effect of merging the multiple candidate propagations before and after the scribbles are requested. Merging afterwards provides finer annotation capabilities as we request scribbles for multiple complementary candidates, which leads to better performance as expected. Interestingly, applying the scribbles without regularization is detrimental when applied after merging. This suggests that the scribbles
are not sufficient as a fixing mechanisms for a single segmentation. Instead, they can be used as a weighting mechanism when merging multiple segmentations.

Using Two Different Brushes
Table 5 compares the two-brushes scenario with the single-brush ones. Brush regularization does not really make sense since the workers directly interact with the segmentation in this scenario. In practice, it does not seem to have a big impact. Both $J$ and $F$ values are slightly lower than the single brush variants but not by a significant amount. On the contrary, the temporal stability is better.

Using Additional Propagation Candidates
Figure 10 shows the evolution of the quality with the increasing number of propagation methods. The methods being additionally used are, in an increasing order: LDOF forward, LDOF backward, DF+SVM+Att, FSEG, BVS, and VPN. We handpicked these combinations of methods such that their complementedness is maximized. The evolution of the boundary accuracy ($F$) is similar and thus only provided in the supplementary material.

Impact of Task Load
One variable of our scribble tasks is the length $P$ of the video sequence to annotate. The longer the sequence, the more work needs to be done. Although the task reward was linearly proportional to the sequence length, humans have a limited budget of attention. Thus we evaluate the amount of work our workers did with respect to the task load (i.e. sequence length). For most of our scribble experiments, the sampling interval $P$ is correlated with the propagation quality, which also leads to a different amount of work required for the intermediate frames. The main exception is the FusionSeg [18] method, and thus we use it to evaluate task load. Table 6 provides the quantitative results.

First, we analyze the total number of brush stroke vertices. While this metric contains the work of outlier scribbles, empty scribbles do not contribute and the amount of single point scribbles is not significant in comparison to the total number of vertices ($<1\%$). Thus it is a reasonable proxy for the amount of work. The total number of vertices is somewhat similar to the average for task loads $P = 10, 5, 3$, whereas $P = 25$ generated about $23\%$ fewer vertices. Beyond the amount of work, we considered the scribble coverage: (1) true positive scribble pixels and (2) false positive ones. True positive pixels are positive scribble pixels that cover propagated segmentation pixels that do not match the ground, whereas false positive pixels wrongly cover segmentation pixels that match the ground truth. The brush size was not regularized (i.e. $f = 1$) so as to match the worker’s point of view. As expected from the number of vertices, the counts for $P = 25$ are also smaller. However, the ratio of true positive to false negative is the largest for $P = 25$. These results hint to the possibility that $P = 25$ was too much of a load for some workers, but they also show that workers who accepted those longer tasks tended to do a more accurate work. Thus the task load may possibly be used as a quality filter.

DISCUSSIONS
We further discuss the key findings of our experiments here.

Best Ensemble Method Settings
Segmentation Ensembles: Negative annotation is an important tool for high-quality segmentation. Workers do use it extensively, and one can use it to discover high-quality segmenters. Clustering is not necessary, but it helps maintain quality when replication counts vary. Higher replication counts help, but three workers were often sufficient to create high-quality segmentations. Review automation is possible, but an oracle reviewer enables higher-quality segmentation.
Table 7. Comparisons between the scores $W$ evaluated by crowd workers and the $J$, $F$, and $T$ metrics, as in Table 1. All values are averages over all sequences of the DAVIS dataset. The score $W$ is defined as the average number of positive evaluations for a frame segmentation.

<table>
<thead>
<tr>
<th>Method</th>
<th>$J$</th>
<th>$F$</th>
<th>$T$</th>
<th>$W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDOF</td>
<td>0.760</td>
<td>0.749</td>
<td>0.250</td>
<td><strong>0.204</strong></td>
</tr>
<tr>
<td>DF+SVM</td>
<td>0.764</td>
<td>0.782</td>
<td><strong>0.250</strong></td>
<td>0.192</td>
</tr>
<tr>
<td>DF+SVM+Att</td>
<td><strong>0.822</strong></td>
<td><strong>0.823</strong></td>
<td>0.317</td>
<td>0.198</td>
</tr>
</tbody>
</table>

Scribble Ensembles: The penalty-based merging strategy $\text{wmaj}$ generates more stable results with little dependency on the brush regularization or the initial segmentation quality. Our scribble annotations are not to be considered as fixing mechanisms, but as localized weighting mechanisms for merging multiple segmentation proposals. Using two brushes might be a good idea, but we do not have conclusive results about it yet. Using more diverse segmentation proposals always helps when possible. While using a high task load leads to lower amounts of work being done, it filters outliers. An optimal load was achieved with sequences of 25 frames in our experiments.

Do Workers Agree with Our Metrics?

Initially, before coming up with our scribble task, we considered the task of deciding which of multiple propagation strategies is the best using the crowd. For a given propagation interval, we asked workers to select, for every propagation method, whether each frame propagation was good or bad after having shown them a selection of good and bad propagation results. The results of that experiment were not necessarily conclusive, but they were interesting in that they seemed to go against our validation metrics and thus we detail these findings here.

The propagation method using deep feature classification tends to produce good results in terms of pixel coverage, but the boundaries exhibit pixelation artifacts as illustrated in Figure 5. In this initial experiments, crowd workers often claimed (to our surprise) that these pixelation artifacts were worse than other results which are pixel-wise less accurate but have smoother boundaries (i.e. the optical flow based methods). This is detailed in Table 7, where we present evaluation scores $W$, measured as the average number of positive evaluations, over all DAVIS sequences propagated with our methods based on optical flow (a bidirectional variant selecting only the best out of both directions) and on deep feature classification (without and with the attention model). In practice, we always use the attention model, but here we present results with and without it to highlight that even without attention model, the propagation with deep features is more accurate according to the three metrics. Workers seemed to focus on the smoothness artifacts (which are stronger without attention model). While the relative preference (higher $W$) of LDOF over both other methods (DF+SVM±Att) is not necessarily significant, the results suggest that the human decision is not well correlated with the current metrics either. This calls for metrics that would match human perception better.

Segmentation Propagation for Video Matting

Our scribble-based propagation ensemble can easily be used for other application scenarios. We demonstrate it with an example of trimap propagation. In movie post-production, a common practice is the cut-out of an object from the background with proper opacity, subsequently composited with a novel background. Among typical techniques used for such composition are green-screen keying for constrained background (hence green-screen), and natural image matting for more demanding cases of unconstrained background (hence natural image). The typical input for natural matting is trimap, which defines the opaque foreground and background regions, as well as the transition regions in-between, as seen in Figure 11. Generating trimaps for every frame in a video is a tedious and time-consuming task [3, 35]. Our ensemble method can be adapted to generate high-quality trimaps quickly for a video given the trimaps for the first and last frames.

We propagate the foreground, which is a conservative region that only includes opaque foreground regions, and the non-background part, which includes the foreground together with the opacity transition regions, separately using the proposed pipeline. We combine the two results in the end to get our final trimaps. In order to show an example of the full application scenario, we generated from these trimaps the foreground layer with semi-opacity properly considered, using information-flow matting [2], and composited them onto a novel background as presented in Figure 11.

CONCLUSION

We introduced two novel crowd-guided ensemble methods that combine multiple inputs from the crowd as well as automated algorithms to generate final segmentations that are better than any of their individual components. First, our results show that we can acquire image segmentations of higher quality by combining the work of multiple individuals. They confirm that negative space annotation is effective for crowdsourced segmentation. They also show that using our scribble-based ensemble method, we can automate the review process of our
oracle to a certain extent. In practice, the oracle may well be required because rotoscoping work is often tied to some artistic decision that requires human validation. On the propagation side, our scribble-based ensembles are promising given their cost efficiency, the quality increase over naive ensembles, and their flexibility. Finally, the web interfaces, tools, and data of our experiments will be made public for future research.

REFERENCES


