# Mixing Crowd and Algorithm Efforts to Segment Objects in Biomedical Images

Danna Gurari[1], Mehrnoosh Sameki[2], Zheng Wu[3], and Margrit Betke[2]

[1] University of Texas at Austin, [2] Boston University, [3] The Mathworks, Inc.

**Abstract.** We examine how to leverage the efforts of crowds and algorithms to find the boundaries of objects in biomedical images (segmentation). We propose a modular framework, SAVE, which generates candidate segmentations and then employs voting to choose among the candidates. This framework supports integration of both computer vision and crowdsourcing modules. We evaluated four implementations of SAVE with different combinations of efforts from crowd workers and algorithms and compared the resulting quality to segmentations created by experts, crowds, and algorithms on 305 biomedical images. Our experiments demonstrate how to produce segmentations more accurate than relying on algorithms or crowd workers alone and comparable (statistically similar) in quality to segmentations created by biomedical experts.

## 1 Introduction

Cost-effective image acquisition and storage technologies are empowering researchers to systematically study biological processes that are invisible to the naked eye. Massive amounts of visual data are collected to, for example, quantify the effects of various cancer drugs [1], model embryonic development [2], and learn how to engineer environments to control cell behavior [3]. A commonality for many of these large-scale analyses is they require a step to demarcate the boundaries of objects in images (segmentation).

Our goal is to learn best practices for segmenting the primary object in each biomedical image (**Figure 1a**). We propose a general-purpose framework, SAVE, that supports users to plug in segmentation annotation and voting quality control methods, where either the crowd or algorithms can be recruited for both tasks. To our knowledge, this is the first work to link computer vision and crowdsourcing methods in a single segmentation framework that generalizes across both communities.

Our work was partially inspired to help individuals identify which algorithms are sufficient to collect accurate segmentations for their image sets. Currently, a user can try a "set of segmentation algorithms, each of which was found to work better in specific scenarios" to find "a method that works well" [1]; e.g., see **Figure 1b,c**. Unfortunately, for non-vision specialists, it can be faster to manually trace boundaries themselves than to risk repeatedly applying different algorithms until finding one to trust (assuming an option exists). We show how to create a crowd voting task for the "best" among multiple segmentation

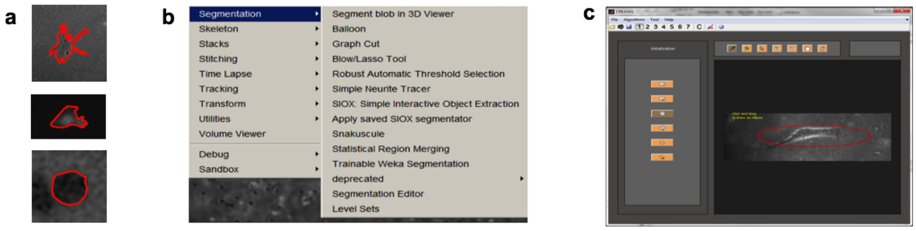


**Fig. 1.** Which segmentation collection method will work best in delineating the boundary of the single primary object in each biomedical image? (**a**) Images exemplify how object appearance can vary significantly with respect to intensity, size, and shape; edges separating objects from the background can be faint; and the backgrounds can be noisy and cluttered. Many bioimage analysis tools include a variety of algorithmic options, such as (**b**) Fiji [14] and (**c**) CREASEG [13].

options per image and then demonstrate its advantage over relying on a single algorithm for all images. While voting for a "best" among multiple options has been adopted for other tasks including classification [4], and detection [5], our work differs by addressing the object segmentation task. This difference necessitates a new user interface design for presenting multiple image-based options as opposed to text-based options.

Our aim to design a crowdsourcing quality control method for object segmentation is shared by prior work [8]. However, prior work employs crowd workers to grade the quality of a *single* crowd-drawn object segmentation. In contrast, we employ crowd workers to identify the best among *multiple* object segmentations, where segmentations can be collected from algorithms or crowds. This is especially valuable for empowering non-algorithm specialists to rapidly deploy the best-suited algorithm for each image, via crowd voting on multiple algorithm-drawn segmentations.

Our work was also partially inspired by the desire to link crowdsourcing and computer vision methods for segmenting objects in biomedical images. Prior work has demonstrated the value of crowd workers from Amazon Mechanical Turk (AMT) [6] and CrowdFlower [7] to reliably segment biomedical images. Our proposed general-purpose framework is advantageous in that control logic in deployed systems can seamlessly shift the load between varying levels of crowdsourcing and algorithm involvement to optimize cost/quality trade-offs.

Finally, our work relates to interactive methods that mix crowd and algorithm efforts to segment images [10–12]. For example, interactive scissors [10] automatically refines crowd worker's annotations as he/she draws. In addition, crowd workers can supply input, as rectangles or coarse segmentations, which are subsequently refined by the Grab Cut algorithm [11]. Another system predicts when to employ humans versus computers to segment an image [12]. Our work grows the limited body of research on hybrid system designs for object segmentation by investigating system workflows that combine crowdsourced lay people and computer vision algorithms for the annotation and quality control steps on a diversity of image content.

## 2 Methods

We propose a modular framework that decomposes the image segmentation task into a series of three micro-tasks. We first describe this framework we call "**S**egmentation **A**nnotation collection, **V**ote Collection, and **E**valuation," or SAVE. We then describe four implementations of this framework that distribute the micro-tasks to algorithms or crowd workers in different combinations. We include a discussion of our new web-based crowd voting tool for the "best" among multiple segmentations that we leverage in the SAVE systems.

**SAVE Framework.** SAVE takes as input an image and outputs a single object segmentation. SAVE involves a series of three steps for each image (**Figure 2**): (1) *Annotation:* $s$ algorithms or humans each annotate the primary object in the image. (2) *Voting:* $n$ votes, at the pixel level or image level, are collected from either humans or an algorithm to determine the "best" annotation from the $s$ annotations (3) *Evaluation:* A decision mechanism interprets the votes to establish a final annotation to use. The key design decisions for implementing this pipeline are to determine (1) which annotation collection methods?, (2) which voters?, and (3) what annotation recommendation decision mechanism?

**Four SAVE Systems.** We implemented four SAVE systems that represent each of the four possible combinations of efforts from crowdsourced workers and algorithms to perform the annotation collection and voting tasks. For each image, our SAVE systems collect five annotations of the image, collect five votes, and then save the segmentation resulting from the majority vote.

*Annotation Collection Implementations.* For crowdsourcing, crowd workers that review our annotation jobs are first presented our five step set of instructions, that include examples of desired and undesired annotations (**Figure 3a**). After the crowd worker accepts our segmentation annotation job, he/she is redirected to the annotation tool (**Figure 3b**). We leverage the freely-available code for LabelMe [15], which performs the widely-adopted approach of sequentially connecting a crowd worker's clicks on an image with straight lines until a closed polygon is completed [11, 8, 7].

For algorithms, we compiled a comprehensive set of 11 options that together span four categories of algorithms commonly reported in the biomedical image segmentation literature [6, 13]. The set consists of thresholding methods (adaptive and Otsu), feature-based methods (Hough Transform and Variance map), a region growing method (watershed), and deformable model based methods (six level set methods from CREASEG [13]).

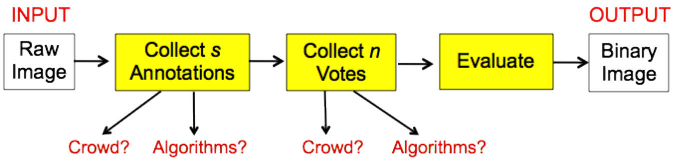
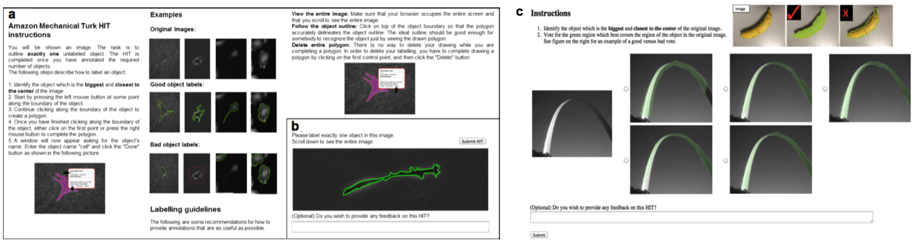


**Fig. 2.** SAVE (Segmentation Annotation collection, Vote collection, and Evaluation) collects $s$ annotations, then collects $n$ votes indicating which pixels/annotations represent the true segmentation, and finally evaluates to establish a final segmentation.



Amazon Mechanical Turk HIT instructions

You will be shown an image. The task is to outline **exactly one** unlabeled object. The HIT is completed once you have annotated the required number of objects.
The following steps describe how to label an object:

1. Identify the object which is the **biggest** and **closest to the center** of the image.
2. Start by pressing the left mouse button at some point along the boundary of the object.
3. Continue clicking along the boundary of the object to create a polygon.
4. Once you have finished clicking along the boundary of the object, either click on the first point or press the right mouse button to complete the polygon.
5. A window will now appear asking for the object's name. Enter the object name "cell" and click the "Done" button as shown in the following picture.

Examples

Original Images:

Good object labels:

Bad object labels:

Labelling guidelines

The following are some recommendations for how to provide annotations that are as useful as possible.

**View the entire image:** Make sure that your browser occupies the entire screen and that you scroll to see the entire image.
**Follow the object outline:** Click on top of the object boundary so that the polygon accurately delineates the object outline. The ideal outline should be good enough for somebody to recognize the object just by seeing the drawn polygon.
**Delete entire polygon:** There is no way to delete your drawing while you are completing a polygon. In order to delete your labelling, you have to complete drawing a polygon by clicking on the first control point, and then click the "Delete" button.

Please label exactly one object in this image.
Scroll down to see the entire image.

(Optional) Do you wish to provide any feedback on this HIT?

Instructions

1. Identify the object which is the **biggest** and **closest to the center** of the original image.
2. Vote for the green region which best covers the region of the object in the original image. See figure on the right for an example of a good versus bad vote.

(Optional) Do you wish to provide any feedback on this HIT?

**Fig. 3.** User interfaces for crowdsourcing annotation and voting tools: (**a**) Instructions we created for crowd workers to use LabelMe to complete the annotation task. (**b**) LabelMe annotation tool used by crowd workers. (**c**) Web-based interface we created for use by crowd workers to vote for a best among multiple segmentations.

*Vote Collection Implementations.* For crowdsourcing, we created a segmentation voting tool which includes instructions followed by the original image on the left and segmentation options on the right (**Figure 3c**). Instructions include exemplar images to demonstrate that the task is to choose the segmentation with the largest number of pixels overlapping the object of interest rather than the segmentation for which the object could be best recognized. We chose to overlay each segmentation option on the original image rather than present the segmentation as a binary image to encourage users to choose the option that is pixel perfect rather than semantically meaningful. We presented all segmentation options in a grid layout consisting of two rows. We scaled all segmentation options to span the maximum width and/or height of their allotted grid cells in the webpage in order to avoid cumbersome user scrolling. The order of segmentation options presented is randomized for each image to prevent biases. To vote, a user selects a radio button next to the desired segmentation and then clicks a submit button.

For algorithms, we implemented pixel level voting to create a final segmentation. Specifically, the algorithm takes as input $N$ segmentations and outputs a single segmentation where a pixel is labeled as foreground when at least $M$ of the segmentations label it as foreground and background otherwise. In other words, in algorithm voting, $N$ pixels cast a vote to indicate the corresponding pixel value in the final image.

*Evaluation Implementation.* We apply a majority vote to determine the final segmentation and, when there is a tie, we select the first segmentation result that accrues the most votes. In general, more stringent levels of voting agreement are advantageous to minimize the chance of mistakes for applications where object shape is critical, such as when shape influences medical diagnoses.

## 3 Experiments and Results

We conducted studies to evaluate and compare the segmentations created by the four proposed SAVE implementations against segmentations created by experts, crowd workers, and algorithms. We examined (1) what is the advantage of our SAVE approach over relying on stand-alone experts, crowd workers, and algorithms? and (2) what should we expect from crowd workers, with respect to skill level and time, for both the annotation and voting tasks?

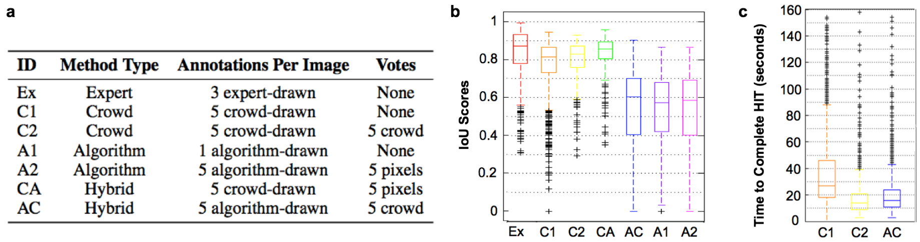
*Datasets.* We conducted our studies on 305 images coming from the publicly-shared BU-BIL [6]. This dataset was created to help scientists with image-based studies on health care problems, such as cancer (melanoma cells) and heart disease (fibroblasts). It includes one magnetic resonance, two fluorescence microscopy, and three phase contrast microscopy datasets. Objects exhibit large variation in scale, extend from highly irregular to circular shapes, and include images with faint edges demarcating the object from the background. The dataset also includes multiple expert segmentations per image with gold standard segmentations that were created by fusing the expert-drawn segmentations.

*Crowdsourcing Platform and Participants.* We recruited crowd workers from Amazon Mechanical Turk (AMT). With this platform, we post jobs (called Human Intelligence Tasks or HITs) paired with a price we paid upon completion of each job. For every HIT, we allotted a maximum of ten minutes to complete the job. We paid \$0.02 for each annotation HIT and \$0.01 for each voting HIT.

*Implementation.* We evaluated four SAVE systems (*C2*, *A2*, *CA*, *AC*) against efforts from crowd workers (*C1*), algorithms (*A1*), and experts (*Ex*) (**Figure 4a**). Two of the approaches are pure crowdsourcing methods: we collected five crowd-drawn annotations (*C1*) and then used the majority vote winner from five crowd votes on the five annotations (*C2*). Another two of the studied approaches are pure automated methods. We chose the overall top-performing algorithm from 11 options (*A1*) and then performed algorithm voting to fuse five of the top-performing algorithms into a single segmentation (*A2*). The next two studied approaches are hybrid algorithm-crowd methods: segmentations created by algorithm voting on the five crowd-drawn segmentations (*CA*) and segmentations chosen from the majority vote from five crowd votes on the aforementioned five algorithm-drawn segmentations (*AC*). The final studied approach is experts (*Ex*) and we used three annotations per image shared with the image library.

**Analysis of Segmentation Quality from Seven Approaches.** We first computed the intersection over union (IoU) score for every segmentation produced by all experts (*Ex*), pure crowd-based (*C1, C2*), pure algorithm-based (*A1, A2*), and hybrid crowd-algorithm based methods (*CA, AC*). The IoU score indicates the similarity of a segmentation to a gold standard segmentation by computing what fraction of pixels are in common to both segmentations (i.e., $\frac{|A \cap B|}{|A \cup B|}$). Scores range from 0 to 1 with higher scores indicating better performance. **Figure 4b** shows the performance of the four SAVE systems compared to the efforts from crowd workers, algorithms, and experts.

*Expert Equivalent Options.* Our top-performing approach of algorithm voting on crowdsourced annotations (*CA*) yields segmentations comparable in quality to those created by experts (*Ex*) ($p < 0.05$, Multiple Comparison Test). Moreover, we observe less variability in segmentation quality with our approach than with experts, as exemplified by smaller inter-quartile ranges (**Figure 4b**). Our findings demonstrate how experts can rely on crowdsourcing to produce segmentations for scientific purposes. At the scale of collecting 10,000 object seg-



| ID | Method Type | Annotations Per Image | Votes |
|---|---|---|---|
| Ex | Expert | 3 expert-drawn | None |
| C1 | Crowd | 5 crowd-drawn | None |
| C2 | Crowd | 5 crowd-drawn | 5 crowd |
| A1 | Algorithm | 1 algorithm-drawn | None |
| A2 | Algorithm | 5 algorithm-drawn | 5 pixels |
| CA | Hybrid | 5 crowd-drawn | 5 pixels |
| AC | Hybrid | 5 algorithm-drawn | 5 crowd |

**Fig. 4.** For 305 biomedical images, we show results with respect to quality and time. (**a**) We describe the seven segmentation approaches we evaluated and compared in the studies. Each segmentation method is described in terms of the SAVE pipeline (i.e., "Annotation" and "Vote" collection). (**b**) Then, we show the box plots indicating all IoU scores for the seven segmentations approaches, where segmentations are evaluated against gold standard segmentations created by fusing multiple expert annotations. We also show (**c**) the time crowd workers took to complete their jobs. For each box in the box plots, the central mark denotes the median value, box edges denote the 25th and 75th percentiles values, whiskers denote the adjacent value to the data point that is greater than one and a half times the size of the inter-quartile range, and black cross-hairs denote outliers. Algorithm voting to fuse crowdsourced drawings (CA) produces expert-quality segmentations ($p < 0.05$).

mentations, our approach would save experts over four forty-hour work weeks of annotating with no loss to quality[1].

*Replacing One-Size-Fits-All Algorithms.* Involving crowd workers to identify which segmentation algorithm to apply from five options (*AC*) outperformed relying on the top-performing option from 11 algorithms (*A1*), as exemplified by the median value improving from 0.57 to 0.6 (**Figure 4b**). Our approach also yielded a 10 percentage point improvement over the best a lay person could achieve today of randomly choosing from the five algorithm options per image; i.e., mean IoU improved from 0.45 to 0.55. Our findings demonstrate how to empower lay people to inexpensively and rapidly collect segmentations at a higher quality than expected even from applications specialists with extensive training about various segmentation algorithms.

It is noteworthy that crowd voting (**AC**) offered an advantage over algorithm voting (**A2**). This is because human selection could accurately find the best option when few algorithm-drawn options were high quality (**Figure 5**, row 2).

*SAVE vs Stand-Alone Crowdsourcing Approach.* Our SAVE systems (*CA*, *C2*) offered significant performance gains over employing standalone annotation collection from crowds (*C1*) ($p < 0.05$, Multiple Comparison Test). Combining multiple crowd-drawn segmentations with crowd and algorithm voting improved overall quality while eliminating most of the egregious outliers (**Figure 4b**).

We found that algorithm voting (**CA**) offered an advantage over crowd voting (**C2**) not only in terms of cost, but also quality ($p < 0.05$, Multiple Comparison

[1] In a user study, we found an expert annotated 423 cells in biomedical images in one eight-hour time period. We use this finding to estimate the time savings.

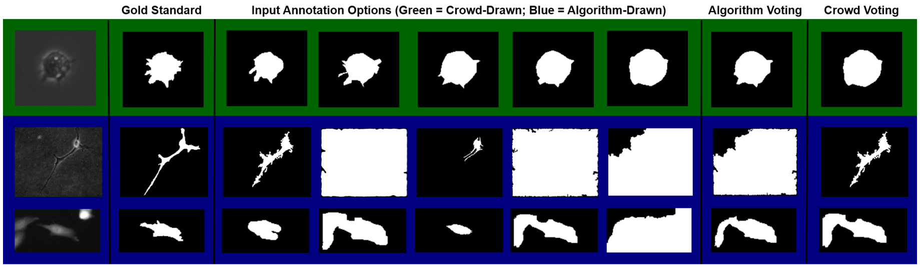

Gold Standard
Input Annotation Options (Green = Crowd-Drawn; Blue = Algorithm-Drawn)
Algorithm Voting
Crowd Voting

**Fig. 5.** Results from our SAVE systems compared to gold standards established from expert annotations. Shown are raw images (col 1), gold standards (col 2), five collected segmentation options (col 3-7), results from algorithm voting (col 8), and results from crowd voting (col 9). As observed, our general framework (SAVE) is fundamentally limited by the quality of segmentation options. If most/all segmentation candidates are comparable, as typical for crowds, algorithm voting is sufficient (row 1). If only a few segmentation candidates are good, as typical for algorithms, human selection is important to identify the best option (row 2). If no good candidate annotations are generated, both voting approaches can be doomed (row 3).

Test). As observed in row 1 of **Figure 5**, algorithm voting can effectively clean up minor mistakes from individual annotators when all options are high quality.

**Characterization of Crowd Behavior.** For all posted HITs, 40 unique crowd workers contributed to the 1,525 segmentations annotations, 45 unique crowd workers completed the 1,525 votes on crowd-drawn annotations, and 44 unique crowd workers completed the 1,525 votes on algorithm-drawn annotations. We analyzed timing results for each batch of HITs coming from the three experiments (i.e., *C1*, *C2*, *AC*) to highlight what one may expect.

*Time Per HIT.* **Figure 4c** shows the time crowd workers took to complete each HIT, as indicated by the lapsed time between when the crowd worker clicked the “Accept” button and “Submit” button for each HIT. Crowd workers spent, on average, approximately twice as much time to complete an annotation task than a voting task; i.e. a median of 27 seconds versus 12 seconds and 16 seconds.

*Elapsed Time Per Experiment.* When examining the elapsed time between posting and completion of all HITs per batch, segmentation voting was completed approximately four times faster than annotation. Specifically, collecting all votes on crowd-drawn segmentations took 88 minutes, all votes on algorithm-drawn segmentations took 62 minutes, and all segmentation annotations took 1,688 minutes. This translates to crowd workers completing 1,040-1,476 votes per hour and 54 annotations per hour. As we expected, our findings demonstrate that users can expect a much quicker turn-around time for experiments relying exclusively on voting than when including annotations. Our SAVE framework is a general purpose pipeline in which deployed *AC* systems could seamlessly shift the annotation load to algorithms to achieve significant time-savings while still typically collecting high quality segmentations.

## 4 Conclusions

We proposed a general-purpose segmentation framework, SAVE, which supports users to interchangeably plug in crowdsourcing and computer vision modules. We quantified segmentation quality resulting from seven pure human-based, pure algorithm-based, and hybrid implementations in order to suggest how to deliver the most effective integration of crowdsourcing and computer vision components in the SAVE framework. We observed the greatest advantages of SAVE systems emerge when combining the strengths of crowdsourcing accuracy and algorithm efficiency. Recruiting crowd workers to vote for the best segmentation from five options (AC) outperformed 11 algorithms and recruiting algorithm voting to fuse five crowdsourced options (CA) produced segmentations comparable in quality to those created by biomedical experts.

## Acknowledgments

The authors thank crowd workers for participating in our experiments and gratefully acknowledge funding from the National Science Foundation (IIS-1421943).

## References

1. D. R. Chittajallu et al. In vivo cell-cycle profiling in xenograft tumors by quantitative intravital microscopy. *Nature Methods*, 12(6): 577–585, 2015.
2. F. Amat et al. Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nature Methods*, 2014. 8 pages.
3. A. S. Hansen and N. Hao and E. K. O'Shea. High-throughput microfluidics to control and measure signaling dynamics in single yeast cells. *Nature Protocols*, 10(8): 1181–1197, 2015.
4. J. Deng and J. Krause and L. Fei-Fei. Fine-Grained Crowdsourcing for Fine-Grained Recognition. *CVPR*, pages 580–587, 2013.
5. K. I. Murray. Multiverse: Crowd Algorithms on Existing Interfaces. *CHI*, pages 2737–2742, 2013.
6. D. Gurari et al. How to Collect Segmentations for Biomedical Images? A Benchmark Evaluating the Performance of Experts, Crowdsourced Non-Experts, and Algorithms. *WACV*, pages 1169–1176, 2015.
7. H. Irshad et al. Crowdsourcing Image Annotation for Nucleus Detection and Segmentation in Computational Pathology: Evaluating Experts, Automated Methods, and the Crowd. *PSB*, 2015. 12 pages.
8. T. Lin et al. Microsoft COCO: Common Objects in Context. *ECCV*, pages 740–755, 2014.
9. S. Bell et al. OPENSURFACES: A Richly Annotated Catalog of Surface Appearance. *TOG*, 32(4), 2013. 11 pages.
10. J. Little and A. Abrams and R. Pless. Tools for richer crowd source image annotations. *WACV*, pages 369–374, 2012.
11. S. Jain and K. Grauman. Predicting Sufficient Annotation Strength for Interactive Foreground Segmentation. *ICCV*, pages 1313–1320, 2013.
12. D. Gurari et al. Pull the Plug? Predicting if Computers or Humans Should Segment Images. *CVPR*, 2016. 10 pages.
13. T. Dietenbeck et al. Creaseg: A free software for the evaluation of image segmentation algorithms based on level-set. *ICIP*, pages 665–668, 2010.
14. J. Schindelin et al. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, page 676–682, 2012.
15. B. C. Russell et al. LabelMe: a database and web-based tool for image annotation. *IJCV*, 77(1-3), pages 157–173, 2008.