BASS: Boundary-Aware Superpixel Segmentation

Antonio Rubio[1,2], LongLong Yu[2], Edgar Simo-Serra[3], Francesc Moreno-Noguer[1]
[1]Institut de Robòtica i Informàtica Industrial (CSIC-UPC), [2]Wide Eyes Technologies, [3]Waseda University
Email: arubio@iri.upc.edu, longyu@wide-eyes.it, esimo@aoni.waseda.jp, fmoreno@iri.upc.edu

***Abstract*—We propose a new superpixel algorithm based on exploiting the boundary information of an image, as objects in images can generally be described by their boundaries. Our proposed approach initially estimates the boundaries and uses them to place superpixel seeds in the areas in which they are more dense. Afterwards, we minimize an energy function in order to expand the seeds into full superpixels. In addition to standard terms such as color consistency and compactness, we propose using the geodesic distance which concentrates small superpixels in regions of the image with more information, while letting larger superpixels cover more homogeneous regions. By both improving the initialization using the boundaries and coherency of the superpixels with geodesic distances, we are able to maintain the coherency of the image structure with fewer superpixels than other approaches. We show the resulting algorithm to yield smaller Variation of Information metrics in seven different datasets while maintaining Undersegmentation Error values similar to the state-of-the-art methods.**

I. Introduction

Representing images as a non-overlapping set of superpixels is a standard practice as a pre-processing step for many computer vision applications, including depth estimation [12], localization [2], tracking [25], gesture recognition [23], human pose estimation [9], place recognition [15] and semantic segmentation [17]. By using superpixels instead of raw pixels, algorithms become more computationally efficient, with the added advantage that superpixels contain much more discriminative information than single pixels, e.g., color histograms and shape.

Superpixels are expected to reduce image complexity while respecting the boundaries, and at the same time they should avoid loss of information due to undersegmentation. The trade-off between these two requirements has been tackled via Normalized Cuts [16], mean shift [4], local variation [8], geometric flows [24], [11] and watershed [22]. Another standard constrain when computing the superpixels consists in homogeneously distributing them along the image and keeping their sizes within limited bounds.

In contrast, we argue that in many situations, the superpixels can be safely merged and their number highly reduced, simplifying thus subsequent tasks. For this purpose, we introduce two main ingredients: 1) we first propose a new approach that spreads the initial superpixels seeds non-uniformly, depending on the image content, and 2) we leverage on image intensity boundaries and a geodesic distance metric to produce smaller superpixels where there is potentially more information in the image (i.e., regions with more intensity boundaries), and bigger superpixels in regions with less presence of boundaries. By doing this, we simultaneously prevent extreme over-segmentation without information gain, and avoid under-segmentation in regions where more precise superpixels are needed. As shown in Fig. 1 and expanded in the results section, our approach brings numerous advantages and improved segmentation metrics compared to the most recent methods[1].

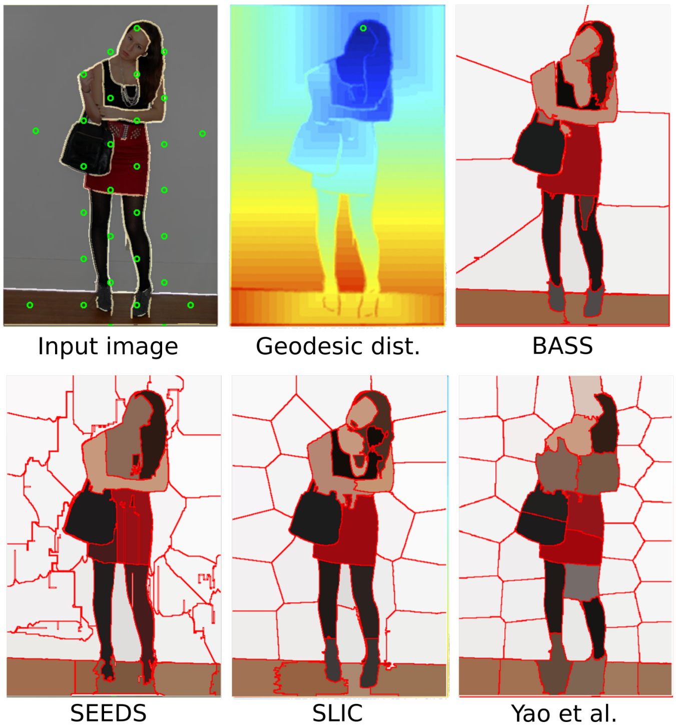
Input image
Geodesic dist.
BASS
SEEDS
SLIC
Yao et al.

Figure 1: Overview of the proposed method. First row (from left to right): input image, with overlaid boundaries and initial seeds positions; geodesic distance with respect to a specific seed; and result of our Boundary-Aware Superpixel Segmentation (BASS) with 26 superpixels. Second row: results of state-of-the-art superpixel segmentations SEEDS [19] (36 superpixels), SLIC [1] (36 superpixels), and Yao et al. [28] (48 superpixels). Even with a smaller number of superpixels, our algorithm is able to achieve better results for the Variation of Information (VOI) metric while maintaining the Undersegmentation Error value when compared with state-of-the-art methods.

In summary, the essential contributions of this paper are:

- A new boundary-aware initialization method for super-

[1] The code will be publicly available in the author's webpage.

pixel centers.

- Use of an energy function that takes into account color information and both Euclidean and geodesic distance between pixels.
- Exhaustive evaluation of the resulting algorithm in seven different datasets (both multiclass and foreground/background) with two different metrics.
- Better Variation of Information metric than state-of-the-art methods and similar value for Undersegmentation Error for a smaller number of superpixels.

II. Related Work

Superpixel computation approaches can be roughly split into three main categories: methods based on graph cuts, techniques that grow superpixels from an initial set of seeds, and techniques that move boundaries from an initial regular grid. We next review each of these families.

A. Graph-based algorithms

Standard approaches use graphs to represent similarities between neighboring pixels, with the pixels being the nodes of the graph, and the edges their similarities. The Normalized Cuts (NC) algorithm [16] may then be used to estimate superpixels by globally minimizing a graph-based objective function. However the computational cost of NC is quite expensive, taking several minutes for segmenting a 480 × 320 pixel image. Other works have proposed alternatives to speed up the process by using agglomerative clustering of the nodes [8], decomposing the graph in multiple scales [5] or by adding grouping constraints [7]. In Graphcut [21], one of the most well known approaches, the constraints for the label of a pixel come from a dense set of overlapped patches, enforcing the regularity of the superpixels.

B. Seed-growing methods

The Watershed method [22] is one of the first non-based graph techniques. It computes superpixels by flooding the gradient image, interpreted as a topological surface. Quick-Shift [20] builds upon the mean-shift algorithm to develop a non-iterative mode-seeking algorithm for clustering. While these algorithms are considerably fast, they produce irregular superpixels which tend to span across different objects. This is improved by the turbopixels algorithm [11], that grows boundary curves from seeds uniformly distributed over the image following geometric flows. The SLIC algorithm [1] is based on the same principle, and substantially improves the efficiency of previous methods. SLIC's main idea is to cluster pixels around regularly distributed seeds based on an energy function that uses both color and Euclidean distance in the image plane. Wang et al. [24] also grow superpixels around regularly distributed seeds, but allows them to split based on the geodesic distance between the pixels and the seeds. All the methods within this category are more efficient than graph-based algorithms, being SLIC the fastest among them. Nonetheless, their performance is not always better. Our method follows this line of work, but we primarily favor reducing the number of superpixels to achieve a certain quality of the segmentation.

C. Coarse-to-fine methods

Another usual choice for superpixel segmentation is to start from a regular grid of superpixels, whose boundaries will iteratively be warped until reaching the termination condition by moving blocks between adjacent superpixels. The size of the *blocks* that move from one superpixel to another is reduced in each iteration until reaching the size of one pixel. The SEEDS [19] algorithm exploits this technique with a simple hill-climbing optimization, using an energy function that enforces color similarity between the boundaries and the superpixel color histogram. Yao et al. [28] uses a similar approach, adding a new topology preserving term to the energy function and focusing on obtaining real-time performance.

While most of the methods in these families focus on producing regular superpixels with similar sizes, we argue that it is convenient to vary the superpixel size in different regions of the image depending on the amount of information present on each region. The goal is to avoid extreme over-segmentation of the image in order to simplify the representation obtained for subsequent applications without deteriorating the quality of the segmentation.

III. Boundary-aware Superpixel Segmentation

Commonly, superpixel algorithms group pixels based on L_2 distance computed in a 5-dimensional space of color and pixel coordinates. In this way, if two pixels are close and have a similar color, they tend to be grouped into the same superpixel.

While this is an standard practice, it ignores the information along the path joining pairs of pixels, which can produce undesirable effects such as undersegmentations. Furthermore, many state-of-the-art algorithms force superpixels to be regular-sized and homogeneously distributed over the image. Again, this seems to be a reasonable heuristic to apply, however, it is prone to produce excessive over-segmentations in regions where small superpixels are unnecessary, such as backgrounds or large regions with homogeneous color.

These methods produce satisfactory results when the number of superpixels is appropriately provided. Nonetheless, in many cases an extreme over-segmentation is needed in order for superpixels to adapt to the ground-truth boundaries. This fact implies a higher cost in the computation of the segmentation. Furthermore, since superpixels are mainly used as a compressed representation for images in higher-level tasks, increasing the number of superpixels also increases the complexity of these applications.

In this paper, we address the problem with the goal of producing more "useful" superpixels, preventing extreme over-segmentation while still producing an accurate representation of the image for subsequent tasks. In order to do that, we compute the boundaries of the image and increase

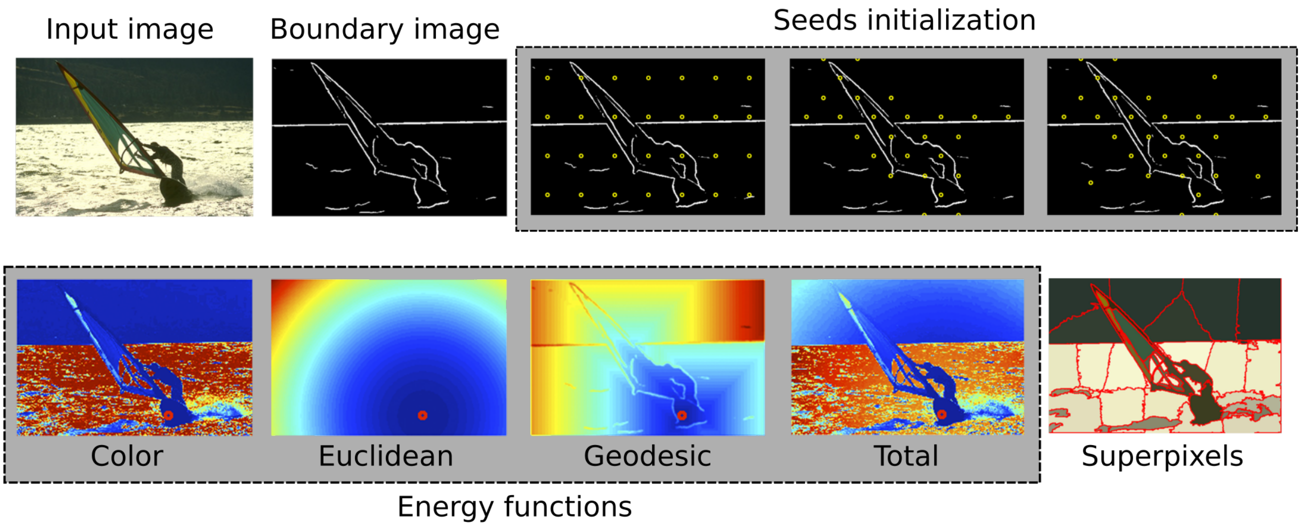

Figure 2: Summary of the main steps of the method. First, the boundary image is obtained. Seeds are regularly distributed over the image, and based on the density of edges, some of them are deleted and some intermediate seeds are added. After that, more seeds are placed in the center of big empty spaces. Once the seeds positions are determined, the method iterates computing the energy function for each seed, and assigning labels to pixels trying to minimize the total energy. Once the termination condition is reached, the connectivity of the labeled pixels is enforced, achieving the final superpixel segmentation.

the concentration of superpixels in regions with more edges, where more detail is necessary. Consequently, superpixels in these regions are smaller than those located in more homogeneous ones (with few edges). Moreover, drawing inspiration in [24], we modify the energy function to be minimized by adding a new term that takes into account the geodesic distance between two points, which helps to retain the structure. Yet, note that [24] does still produce quite homogeneous superpixels, not content aware sized superpixels as we do.

We next describe the steps of the algorithm we propose. Refer to Fig. 2 for a visual explanation.

A. Boundary detection

For each input image we compute its boundary image using an off-the-shelf structured forest-based approach [6], which has been proven to run in real-time while providing state-of-the-art results in the *BSDS500* dataset [3] To simplify the computation of geodesic distances we binarize the edge detection result, using only the top 70% most intense boundaries.

B. Seeds initialization

Unlike other seed-based state-of-the-art algorithms (such as SLIC [1]), that regularly distribute the seeds over the image, we place more seeds for superpixels in regions with large boundary concentration. This is done in three steps as outlined in Fig. 2. Initially, we place seeds following a regular grid spaced S pixels apart ($S = \sqrt{N/K}$, with N the number of pixels of the image and K the desired number of superpixels). After that, based on the ratio of boundary pixels found inside a square region sized $S \times S$ around each seed, we decide whether or not to add or delete any seed by comparison against a certain threshold $T_{ad} = \left(\Sigma_{e_i}\right)/N$, being e_i a pixel in the boundary image (with value 0 or 1), and N the total number of pixels in the image. More formally, the seed addition/deletion operation can be written as:

$$\begin{cases} \text{Add}, & \text{if } (\Sigma_S\, e_i)/N \; > \; 3 \cdot T_{ad} \\ \text{Delete}, & \text{if } (\Sigma_S\, e_i)/N \; < \; T_{ad} \end{cases} \tag{1}$$

where $\sum_S e_i$ represents the sum of all the pixels in the mentioned square region centered in a seed. If the condition for adding seeds is satisfied, four new seeds are created in the corners of such region. Note that the condition for adding is harder than that for deleting, as our objective is minimizing the final number of superpixels while maintaining a good quality in the segmentation. Finally, we place a seed in the mass center of empty regions with areas larger than $S \times S$ pixels.

C. Energy function

The label assignation consists of an iterative clustering based on an energy function E composed of three terms associated to color information and Euclidean and geodesic distances. Previously defined seeds act like cluster centers in a 5-dimensional k-means problem:

$$S_k = \left[l_k, a_k, b_k, x_k, y_k\right]^T \tag{2}$$

where $\left(x_k, y_k\right)$ are the pixel coordinates of seed S_k on the image and (l_k, a_k, b_k) are its color values in CIELAB color space. Each seed has an associated label.

The optimization process consists of several iterations over all seeds, computing an energy value for their surrounding pixels and assigning them the label of the seed that minimizes their energy. At the end of every iteration, the seeds are updated as the mean of the positions and colors of all the pixels that belong to them.

More specifically, at each iteration we compute the total energy for every pixel in a region around each seed as the sum of E_{lab}, E_{xy} and E_{geo}, weighting the two last terms with parameters α and β.

$$E = E_{lab} + \alpha \cdot E_{xy} + \beta \cdot E_{geo} \tag{3}$$

where $\alpha = {}^{C}/_{S}$, being C a compactness term and S the already defined *step*. The two first energy terms, corresponding to color and Euclidean distance, are computed as in [1]:

$$E_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{4}$$

$$E_{xy} = \sqrt{(x_k - x_i)^2 + \left(y_k - y_i\right)^2} \tag{5}$$

The last energy term we propose depends on the gray-weighted geodesic distance computed over the binary boundary image. This distance is defined as the smallest weighted sum of gray levels along the discrete path between two given pixels. Concretely, we implement the Distance Transform on Curved Space from [18]. This operation yields an image where every pixel i has a value corresponding to the distance of that pixel to the nearest seed S_k. The region in which we compute this energy for each seed is sized $2S \times 2S$.

We initialize the energy of all pixels to E_0. A reasonable choice would be to set $E_0 = \infty$, but that would force all pixels to get a label in the first iteration, even when they are not specially close to any seed. For that reason, we set E_0 as a finite value that we linearly increase with the number of deleted seeds. Thus, if the energy of a pixel is not lower than E_0 it will have label $l = 0$, and all pixels with such label will form a superpixel. We then iterate until the maximum allowed number of iterations is reached. After all these steps, we remove those superpixels whose area is too small by merging them with adjacent bigger superpixels

IV. Experimental evaluation

Next, we show the results obtained applying our superpixel algorithm to seven different datasets: *Fashionista* [26], *Berkeley Segmentation Dataset (BSD)* [14], *HorseSeg* [10], *DogSeg* [10], *MSRA Salient Object Database* [13], *Complex Scene Saliency Dataset (CSSD)* and *Extended CSSD (ECSSD)* [27]. *Fashionista* is a multi-class fashion dataset where the model is centered on the image, while *BSD* is also multi-class, but contains all types of images. The rest of datasets have binary segmentations (foreground/background): *DogSeg* and *HorseSeg* are composed of images of dogs and horses collected from ImageNet and PASCAL VOC12. *MSRA* has very different images, but most are both smooth and simple. On the other hand, images from *CSSD* and *ECSSD* present more natural situations.

We compare our results against three state-of-the-art algorithms: SEEDS [19], SLIC [1], and Yao et al. [28]. All algorithms were evaluated with the code from the authors' websites. For BASS, the maximum number of iterations has been experimentally determined as 10 to produce fast segmentations without excessively affecting their quality. A brief description of the metrics used to evaluate the segmentations is given below, followed by a discussion of the results obtained.

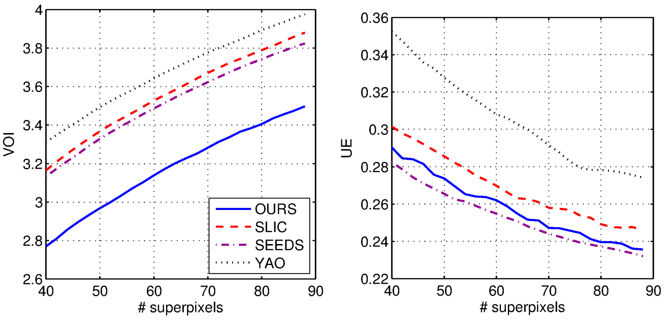

Figure 3: Values of the evaluation metrics for different number of superpixels. As we see, our method outperforms state-of-the-art methods in Variation of Information metric and obtains the second best result in Undersegmentation Error. In both cases, lower values correspond to better segmentations.

A. Evaluation Metrics

Variation of Information (VOI). It measures the distance between two different clusterings. Given two segmentations of the same image: $X = \{X_1, X_2, \ldots, X_k\}$ and $Y = \{Y_1, Y_2, \ldots, Y_l\}$, where X_i and Y_j are the superpixels for each segmentation, and n is the total number of image pixels ($n = \sum_i |X_i| = \sum_j |Y_j| = |A|$), VOI is computed as

$$VOI(X;Y) = -\sum_{i,j} r_{ij} \cdot \left[\log\left(\frac{r_{ij}}{p_i}\right) + \log\left(\frac{r_{ij}}{q_j}\right)\right] \tag{6}$$

where $p_i = {}^{|X_i|}/_{n}$, $q_j = {}^{|Y_j|}/_{n}$ and $r_{ij} = {}^{|X_i| \cap |Y_j|}/_{n}$. Lower values correspond to smaller distances and hence to more similar segmentations.

Undersegmentation Error (UE). It is computed as

$$UE = \frac{1}{GT} \sum_{S \in GT} \left(\frac{\sum_{P: P \cap S \neq 0} \min\left(|P_{in}|, |P_{out}|\right)}{|S|} \right) \tag{7}$$

where GT is the number of ground truth segments, P are the superpixel segments, S the ground truth segments, and $|P_{in}|$ and $|P_{out}|$ represent the area of P inside and outside S, respectively. A low value is desirable.

B. Comparison against State of the Art

Since we consider a large number the datasets, the results we next present are computed on 10% of randomly chosen images for each dataset (about 300 images per dataset). This already gives a good intuition of the performance of all algorithms. Note that the number of initial seeds or desired superpixels does not normally coincide with the exact final number of superpixels, so in order to perform a

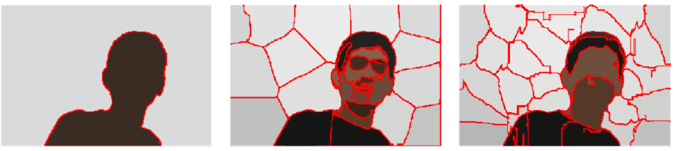

Figure 4: Two segmentations with similar UE (BASS: 0.0077, SEEDS: 0.0112). The segmentation with BASS (with a VOI of 2.5340, lower than the value for SEEDS 2.8095) contains the same information with less superpixels.

fair evaluation, we processed all images with a wide range of initial seeds. In this way, we obtain values for a sufficient variety of actual superpixels for all images to compare.

Figure 3 reports the previous metrics for different number of superpixels, averaged over all seven datasets (the results were quite similar for every dataset). A unique set of parameter values (empirically determined) was used for all the datasets in order to perform fair comparisons and emphasize the generalization of the method, even though specific parameter sets per dataset would give better individual results. These results show how our algorithm consistently decreases the VOI for all number of superpixels and, at the same time, maintains UE values similar to state-of-the-art methods. Indeed, we argue that lower VOI is much more representative for our primary goal of retaining the image information with a minimal number of superpixels. This is clearly illustrated in Figure 4.

C. Qualitative Results

Several images segmented with different numbers of initial seeds for all the methods are shown in Fig. 5. Note how small superpixels are concentrated in more meaningful areas, and how our method is able to capture the same information than the rest while drastically reducing the number of “useless” superpixels, obtaining simpler representations of the images.

V. Conclusions

We have presented an over-segmentation algorithm to compute superpixels that are aware of the boundary information of the input image in order to simplify the final result. We have formulated the problem as a clustering problem using color, Euclidean distance and geodesic distance over an edge image. We evaluate our method against the state-of-the-art using seven different datasets. Our algorithm outperforms state-of-the-art methods in the most significant metric according to our goal while maintaining the quality of the segmentation. The algorithm is implemented in C++ and runs on CPU in about 0.5 seconds per image. We will make our code publicly available.

VI. Acknowledgments

This work is partly funded by the Spanish MINECO project RobInstruct TIN2014-58178-R, by the ERA-Net Chistera project I-DRESS PCIN-2015-147 and by the EU project AEROARMS H2020-ICT-2014-1-644271. A.Rubio is supported by the industrial doctorate grant 2015-DI-010 of the AGAUR.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012.

[2] S. Akbar, L. Jordan, A. M. Thompson, and S. J. McKenna. Tumor localization in tissue microarrays using rotation invariant superpixel pyramids. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, pages 1292–1295. IEEE, 2015.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[5] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

[6] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[7] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. *Journal of Mathematical Imaging and Vision*, 39(1):45–61, 2011.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[9] H. Kim, S. Lee, D. Lee, S. Choi, J. Ju, and H. Myung. Real-time human pose estimation and gesture recognition from depth images using superpixels and svm classifier. *Sensors*, 15(6):12410–12427, 2015.

[10] A. Kolesnikov, M. Guillaumin, V. Ferrari, and C. H. Lampert. Closed-form approximate crf training for scalable image segmentation. In *ECCV*. 2014.

[11] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *PAMI*, 31(12):2290–2297, 2009.

[12] F. Liu, C. Shen, G. Lin, and I. Reid. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

[13] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *PAMI*, 33(2):353–367, 2011.

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[15] P. Neubert, N. Sünderhauf, and P. Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *Robotics and Autonomous Systems*, 69:15–27, 2015.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[17] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. A High Performance CRF Model for Clothes Parsing. In *ACCV*, 2014.

[18] P. J. Toivanen. New geodosic distance transforms for gray-scale images. *Pattern Recognition Letters*, 17(5):437–450, 1996.

[19] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. *IJCV*, 111(3):298–314, 2015.

[20] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *ECCV*. 2008.

[21] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. In *ECCV*. 2010.

[22] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *PAMI*, (6):583–598, 1991.

[23] C. Wang, Z. Liu, and S.-C. Chan. Superpixel-based hand gesture recognition with kinect depth camera. *Multimedia, IEEE Transactions on*, 17(1):29–39, 2015.

[24] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha. Structure-sensitive superpixels via geodesic distance. *IJCV*, 103(1):1–21, 2013.

[25] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011.

[26] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.

[27] Q. Yan, J. Shi, L. Xu, and J. Jia. Hierarchical saliency detection on extended cssd. *arXiv preprint arXiv:1408.5418*, 2014.

[28] J. Yao, M. Boben, S. Fidler, and R. Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *CVPR*, 2015.

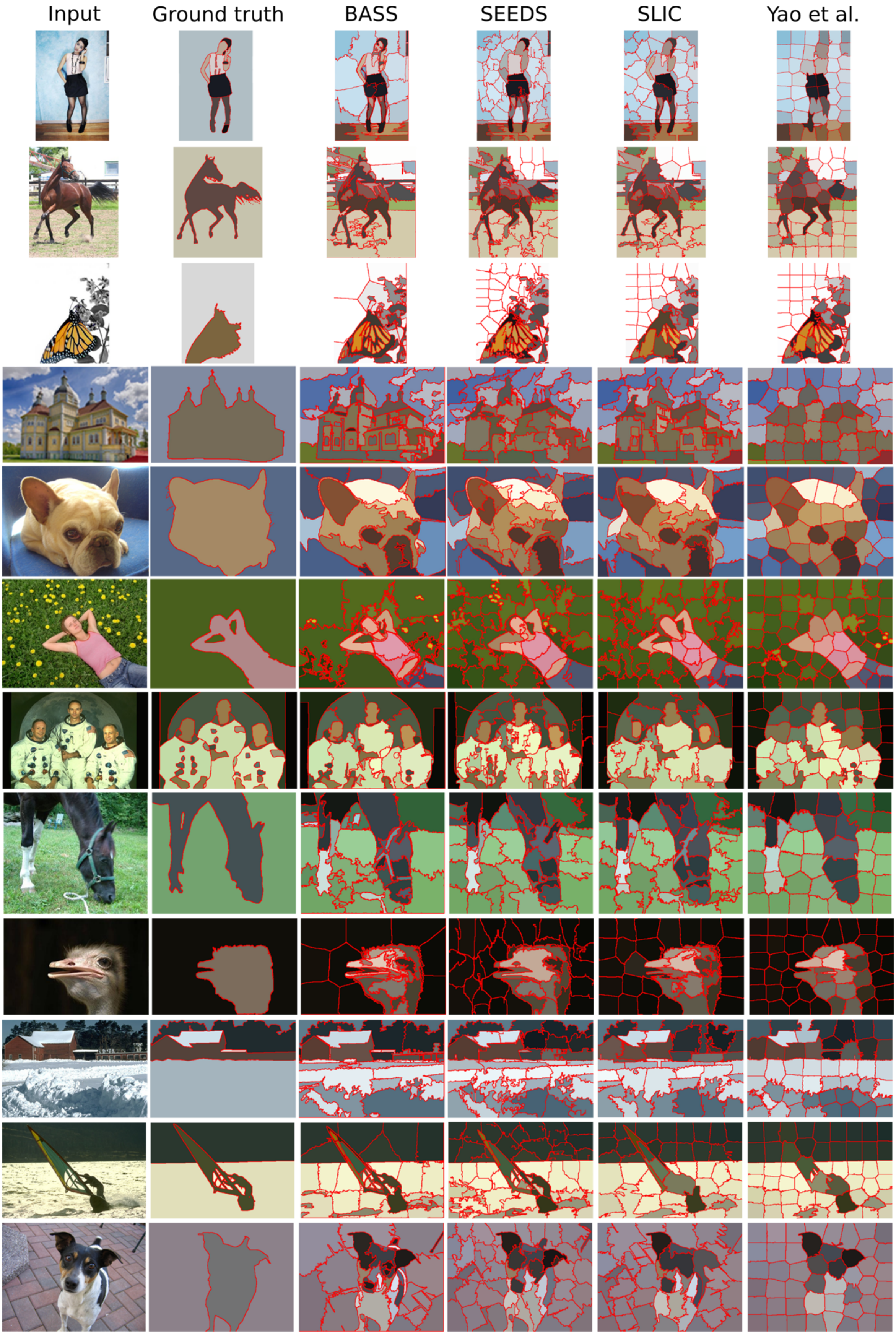

Figure 5: Some results of our superpixel segmentation algorithm compared to state-of-the-art methods.