Object Detection and Segmentation from Joint Embedding of Parts and Pixels

Michael Maire[1], Stella X. Yu[2], and Pietro Perona[1]
[1]California Institute of Technology - Pasadena, CA 91125
[2]Boston College - Chestnut Hill, MA 02467
{mmaire,perona}@caltech.edu, syu@cs.bc.edu

Abstract

We present a new framework in which image segmentation, figure/ground organization, and object detection all appear as the result of solving a single grouping problem. This framework serves as a perceptual organization stage that integrates information from low-level image cues with that of high-level part detectors. Pixels and parts each appear as nodes in a graph whose edges encode both affinity and ordering relationships. We derive a generalized eigenproblem from this graph and read off an interpretation of the image from the solution eigenvectors. Combining an off-the-shelf top-down part-based person detector with our low-level cues and grouping formulation, we demonstrate improvements to object detection and segmentation.

1. Introduction

Many high-performance object detection algorithms operate top-down and do not exploit grouping or segmentation processes. The best algorithms [9, 26] in the PASCAL VOC challenge [8] fall into this category as do top systems for important applications such as finding people in images [2] and detecting pedestrians specifically [6, 7, 22]. When object segmentation is desired as an output, it is often obtained in a post-processing step, for example, by aligning the predictions of a top-down detector to image contours [2].

Proponents of using segmentation as an initial phase for detection and recognition argue that it offers many advantages, such as a reduction in computational complexity [13] or context over which to compute features [20, 17]. A common theme of such work is to first partition the image into a discrete set of intermediate units, such as superpixels [23], regions, or contours [25]. These entities can then either serve as input to object detectors [13, 25], or be reasoned about in concert with detectors to construct a scene interpretation [14, 12, 27, 16].

A drawback of this approach is that it can be difficult to recover from errors introduced in the initial set of regions or contours. Hence, the technique of using multiple segmentations has emerged as a popular method to ameliorate these difficulties [20, 11, 15]. However, such a strategy comes at the cost of increased complexity and still offers no guarantee that the correct partitioning will be available.

This paper explores an alternative to the prevalent trends of either ignoring segmentation or placing a clear division between segmentation and detection. We avoid making any hard decisions on an initial image segmentation, instead integrating low-level segmentation cues into the object detection process in a soft manner. In our framework, segmentation and object detection emerge as two aspects of the same grouping problem.

Our work is in the spirit of prior efforts at using spectral graph theory to optimize joint grouping criteria for pixels and objects [29, 30]. It builds on these ideas, with the following important contributions:

- We use Angular Embedding (AE) [28] in place of Normalized Cuts [24] as the grouping algorithm and take advantage of its additional expressive power. AE was previously used for brightness modeling [28] and figure/ground organization [18]. We further extend its domain of applicability to object detection.
- Unlike [29, 30] where parts are patches specific to individual object views and appearances, we use part detectors specific to object pose [2]. In this setting, we define part-part and part-pixel interactions differently.
- Figure/ground organization, image segmentation, detection of multiple object instances, and construction of per-object segmentation masks occur as consequences of solving a single generalized eigenproblem. All of these quantities are recovered from a new representation of pixels and parts in the embedding space defined by the solution eigenvectors.

Section 2 presents our algorithm, describing the coupling between pixel and part layers, formulation of the optimization problem, and interpretation of the solution eigenvectors. Section 3 provides experimental results on the PASCAL VOC person segmentation task. Section 4 concludes.

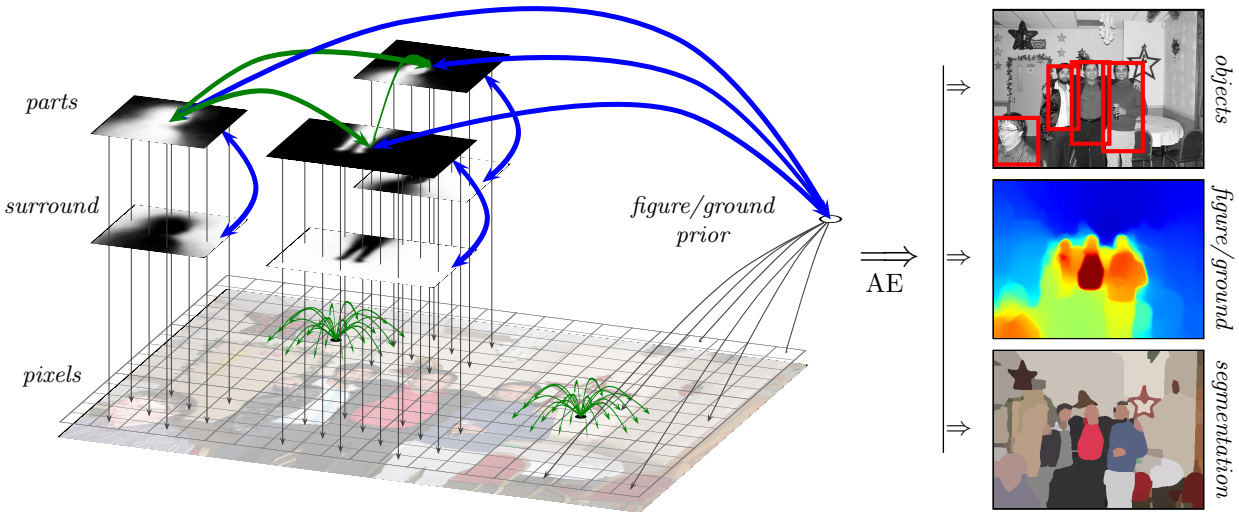

Figure 1. **System diagram.** We construct a graph containing a node for each pixel and two nodes (part and surround regions) for each part detection. Pixels connect to neighboring pixels within a fixed radius with affinity given by the intervening contour cue [10]. Part nodes connect to one another with affinity determined by their agreement on object pose. Each part node is pushed away from its corresponding surround node to enforce figure/ground separation. A dummy node connected to all parts and all pixels provides a weak prior that pushes pixels into the background. Learned figure/ground masks for the parts [3] place requirements on agreement with the pixels they cover. The diagram above displays a subset of the connections, using green arrows for affinity, blue arrows for ordering relationships, and gray arrows for agreement requirements. We solve an Angular Embedding (AE) problem [28] which captures all of these interactions, and obtain a segmentation of the image, a global figure/ground ordering on pixels, and a clustering of parts into detected objects.

2. Grouping Framework

In our system, low-level cues and high-level part detectors work together to organize the image. Figure 1 illustrates the sources of information at play:

- Low-level cues bind pixels to one another, encouraging them to respect region coherence.
- Part detections bind to each other according to their compatibility of belonging to the same object.
- Parts pull the image region they cover into the foreground and push the surrounding area into the background, providing a coarse figure/ground signal.

The rest of this section describes a method for integrating these information sources to reach a global decision about pixel and part grouping, a process we refer to as globalization. While the integration framework depends only on the relationship types, we also present a concrete implementation with cues that deliver good results in practice.

2.1. Globalization

Figure/ground interactions are of a fundamentally different type than grouping relations between pixels or parts. The portion of the image that should be considered figure may be the union of multiple distinct objects and regions. Objects can also occlude one another, suggesting a continuous measure of figure/ground ordering is most appropriate in capturing this aspect of scene layout. We use the recently introduced Angular Embedding (AE) algorithm [28] as a globalization framework, since it has the expressive power to incorporate both affinity and ordering relationships.

A pair (C, Θ) of real-valued matrices captures pixel-pixel and part-part interactions and defines the input to an AE problem. Skew-symmetric matrix Θ specifies relative *ordering* relationships. Symmetric matrix C specifies a *confidence* on each of these relationships. We encode pairwise affinity by setting relative ordering to zero and setting confidence according to the degree of attraction. Figure/ground relationships utilize nonzero relative ordering terms [18].

The output of AE is a representation of both pixels and parts in a complex number space. Distance in this space reflects the notion of grouping, while the phases of the complex numbers encode a global ordering. We capture pixel-part relationships by imposing requirements on the solution space of the embedding [31]. These requirements take the form of a sparse matrix U whose columns specify linear constraints involving pixels and parts.

Given C, Θ, and U, we solve for the complex eigenvectors, $z_0, ..., z_{m-1}$, corresponding to the m largest eigenvalues, $\lambda_0, ..., \lambda_{m-1}$, of the constrained AE problem:

$$QPQz = \lambda z \tag{1}$$

where P is a normalized weight matrix and Q is a projector

onto the feasible solution space:

$$P = D^{-1}W \tag{2}$$

$$Q = I - D^{-1}U(U^T D^{-1}U)^{-1}U^T \tag{3}$$

with D and W defined in terms of C and Θ by:

$$D = Diag(C1_n) \tag{4}$$

$$W = C \bullet e^{i\Theta} \tag{5}$$

where n is the number of nodes, 1_n is a column vector of ones, I is the identity matrix, $Diag(\cdot)$ is a matrix with its vector argument on the main diagonal, $\bullet$ denotes the matrix Hadamard product, $i = \sqrt{-1}$ and exponentiation acts element-wise.

Following the procedure for constrained Normalized Cuts [31], but with complex-valued matrices, (1) can be solved efficiently and without explicit computation of Q by modifying the inner loop of an eigensolver.

We defer interpretation of the resulting eigenvectors to Section 2.4 and first describe how to construct C, Θ, and U.

2.2. Node Relationships

We encode relationships between $n = n_p + 2n_q + 1$ nodes, where n_p and n_q are the number of image pixels and part detections, respectively. Denote pixels as p_i and parts as q_i. For each part q_i, we create an additional node s_i representing its local surround. An extra node f enforces a figure/ground prior through its connections to both pixels and parts. C and Θ are $n \times n$ matrices with block structure:

$$C = \begin{array}{c} \overbrace{}^{n_p}\ \overbrace{}^{n_q}\ \overbrace{}^{n_q}\ \overbrace{}^{1} \\ \begin{bmatrix} C_p & 0 & 0 & 0 \\ 0 & \alpha \cdot C_q & \beta \cdot C_s & \gamma \cdot C_f \\ 0 & \beta \cdot C_s^T & 0 & 0 \\ 0 & \gamma \cdot C_f^T & 0 & 0 \end{bmatrix} \end{array} \tag{6}$$

$$\Theta = \Sigma^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \Theta_s & \Theta_f \\ 0 & -\Theta_s^T & 0 & 0 \\ 0 & -\Theta_f^T & 0 & 0 \end{bmatrix} \tag{7}$$

where C_p stores affinity between pixels, C_q stores affinity between parts, (C_s, Θ_s) encodes separation between part and surround, and (C_f, Θ_f) encodes the figure/ground prior. Weights α, β, and γ trade off the relative importance of the relationship types during globalization. Σ is a normalization factor involving the sum of the absolute values of the entries of Θ_s and Θ_f:

$$\Sigma = \frac{2}{\pi} \cdot \left(1_{n_q}^T |\Theta_s| 1_{n_q} + 1_{n_q}^T |\Theta_f| \right) \tag{8}$$

Scaling by Σ^{-1} guarantees that when embedding into the unit circle of the complex plane to find a global ordering (Section 2.4.1), the angular span of the optimal solution does not exceed π. This scaling removes the potential wrap-around effect in circular embedding.

2.2.1 Pixel Layer

As in recent effective image segmentation approaches based on spectral clustering [5, 19, 1], we use the *intervening contour* cue [10] to define affinities between pixels. We follow the implementation in [19] of computing intervening contour on top of a multiscale version of the *Pb (probability of boundary)* edge detector [21]. Specifically, for pixels p_i and p_j within a fixed radius of one another:

$$C_p(p_i, p_j) = \exp\left(- \max_{x \in \overline{p_i p_j}} \{Pb(x)\} / \rho \right) \tag{9}$$

where $\overline{p_i p_j}$ is the line segment connecting p_i and p_j in the image plane, and ρ is a constant. C_p is sparse as there are no connections over distances larger than the given radius.

2.2.2 Part Layer

We take the publicly available poselet-based body part detector of Bourdev and Malik [3] as our source of top-down parts. Poselets are predictive of both object layout and local figure/ground.

We use a part affinity function motivated by previous work. In particular, Bourdev *et al*. [2] define a distance between poselet activations, which they subsequently use in an ad-hoc greedy iterative clustering procedure. We use the same distance metric, but convert it into an affinity so that poselet grouping becomes one aspect of our globalization process. For parts (or poselets) q_i and q_j:

$$C_q(q_i, q_j) = e^{-\frac{D_{SKL}(q_i, q_j)}{\tau}} e^{\frac{\min\{S(q_i), S(q_j)\} - 1}{\upsilon}} \tag{10}$$

where τ, υ are constants, D_{SKL} is the symmetrized KL divergence of the poselet keypoint distributions $\mathcal{N}_i^k$ and $\mathcal{N}_j^k$:

$$D_{SKL}(q_i, q_j) \propto \sum_k \left[D_{KL}(\mathcal{N}_i^k \| \mathcal{N}_j^k) + D_{KL}(\mathcal{N}_j^k \| \mathcal{N}_i^k) \right] \tag{11}$$

The second factor in C_q scales the affinity according to the part detector scores $S(q_i) \in [0, 1]$ in order to discount weak detections. C_q is dense, but C is still sparse since $n_q \ll n_p$.

We establish a local figure/ground ordering for each object part q_i by connecting it to its corresponding surround node s_i with confidence and ordering given by:

$$C_s(q_i, s_i) = e^{\frac{S(q_i) - 1}{\upsilon}} \tag{12}$$

$$\Theta_s(q_i, s_i) = 1 \tag{13}$$

Separation between q_i and s_i is only meaningful given their connections to pixel grouping, which we discuss next.

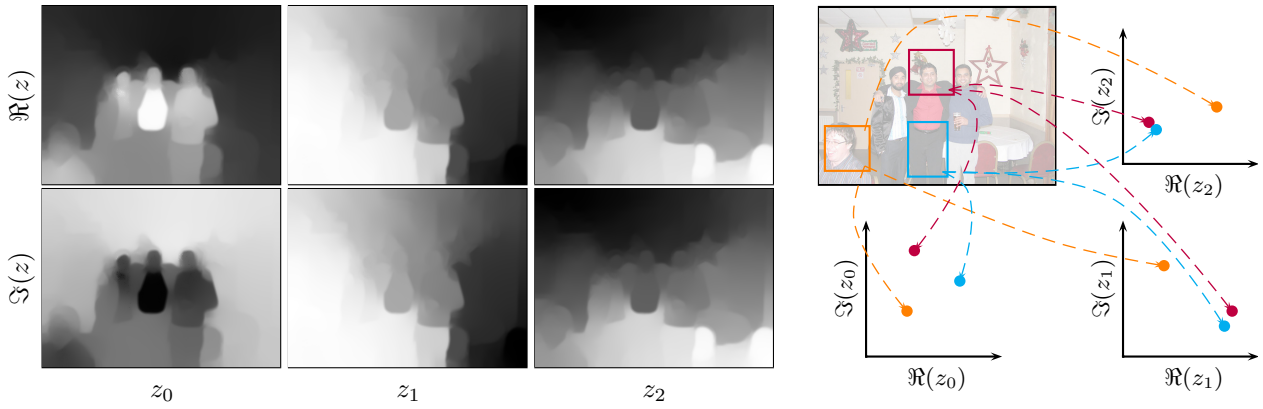

Figure 2. **Eigenvectors carry ordering and clustering information.** The leading m eigenvectors, obtained by solving an Angular Embedding problem that captures our grouping relationships, provide a mapping of graph nodes (pixels, parts, and surround) into $\mathbb{C}^m$. *Left:* Shown are the first three complex eigenvectors, z_0, z_1, z_2, for the pixel nodes of the example from Figure 1. Ordering is captured by $\angle z_0$ (visible in the center panel on the right in Figure 1), which pops out figure from ground. The remaining eigenvectors encode clustering and vary slowly within coherent regions. *Right:* The same is true for the detected parts, with clustering in $\mathbb{C}^m$ determining object membership. Similarity between parts and pixels is also encoded by their proximity in $\mathbb{C}^m$ and permits extraction of object segmentation masks.

2.2.3 Pixel-Part Constraints

We encode part-pixel interactions as constraints on the embedding solution. Each part node must take the average value of its member pixel nodes, so that the part and pixel representations are always consistent. For part node q_i and corresponding surround node s_i:

$$z(q_i) \propto \sum_{p_j \in M_i} \lceil F_i(p_j) \rceil e^{\frac{|F_i(p_j)|-1}{\sigma}} e^{-\frac{Pb(p_j)}{\rho}} z(p_j) \quad (14)$$

$$z(s_i) \propto \sum_{p_j \in M_i} -\lfloor F_i(p_j) \rfloor e^{\frac{|F_i(p_j)|-1}{\sigma}} e^{-\frac{Pb(p_j)}{\rho}} z(p_j) \quad (15)$$

where M_i is the set of pixels overlapping the mask for detection q_i, $F_i(\cdot) \rightarrow (-1, 1)$ is the local figure/ground prediction made by this mask, and σ is a constant. The first term within each sum selects a pixel's membership in either part or surround. The remaining terms weight its contribution by the mask prediction and chance it belongs to a region interior rather than a boundary. We write all of these constraints in matrix form as $U^T z = 0$. For computational efficiency, we increase the sparsity of U by sampling a random subset of image pixels to participate in the constraints.

2.2.4 Figure/Ground Prior

Image regions in which no parts fire are likely to be background, a fact not captured by the local part-surround relationships. To remedy this, we add an extra node f to act as a weak figure/ground prior. We set f to be a weighted average of all pixels (or a sampled subset of them):

$$z(f) \propto \sum_{p_j} e^{-\frac{Pb(p_j)}{\rho}} z(p_j) \quad (16)$$

This constraint tacks another column onto U. Node f then acts as surround with respect to each part q_i:

$$C_f(q_i) = e^{\frac{S(q_i)-1}{\upsilon}} \quad (17)$$

$$\Theta_f(q_i) = 1 \quad (18)$$

In absence of other evidence, placing pixels into the background shifts the location of $z(f)$ and better satisfies these ordering preferences.

2.3. Parameters

Two sets of parameters control our grouping algorithm: $(\rho, \sigma, \tau, \upsilon)$ govern affinities and (α, β, γ) control relative importance of subproblems. We set $\rho = \sigma = \upsilon = 0.1$, interpreting numerators in their respective affinities as probabilities. We set τ appropriately with respect to D_{SKL}. One could learn (α, β, γ) by stochastic gradient descent on a validation set. This is computationally expensive and we believe the system is not too sensitive to these parameters, so we instead set them manually.

2.4. Decoding Eigenvectors

The complex eigenvectors, $z_0, z_1, ..., z_{m-1}$, of Equation (1) corresponding to the m largest eigenvalues define an embedding of the graph nodes into $\mathbb{C}^m$. Figure 2 displays an example. By design of the optimization problem, the locations of the nodes in $\mathbb{C}^m$ are meaningful, in terms of both ordering (given by z_0) and clustering (given by $z_1, ..., z_{m-1}$). Figure 3 illustrates that applying simple transformations to the eigenvector representation allows us to "decode" solutions to our ordering (figure/ground) and clustering (segmentation and object detection) problems.

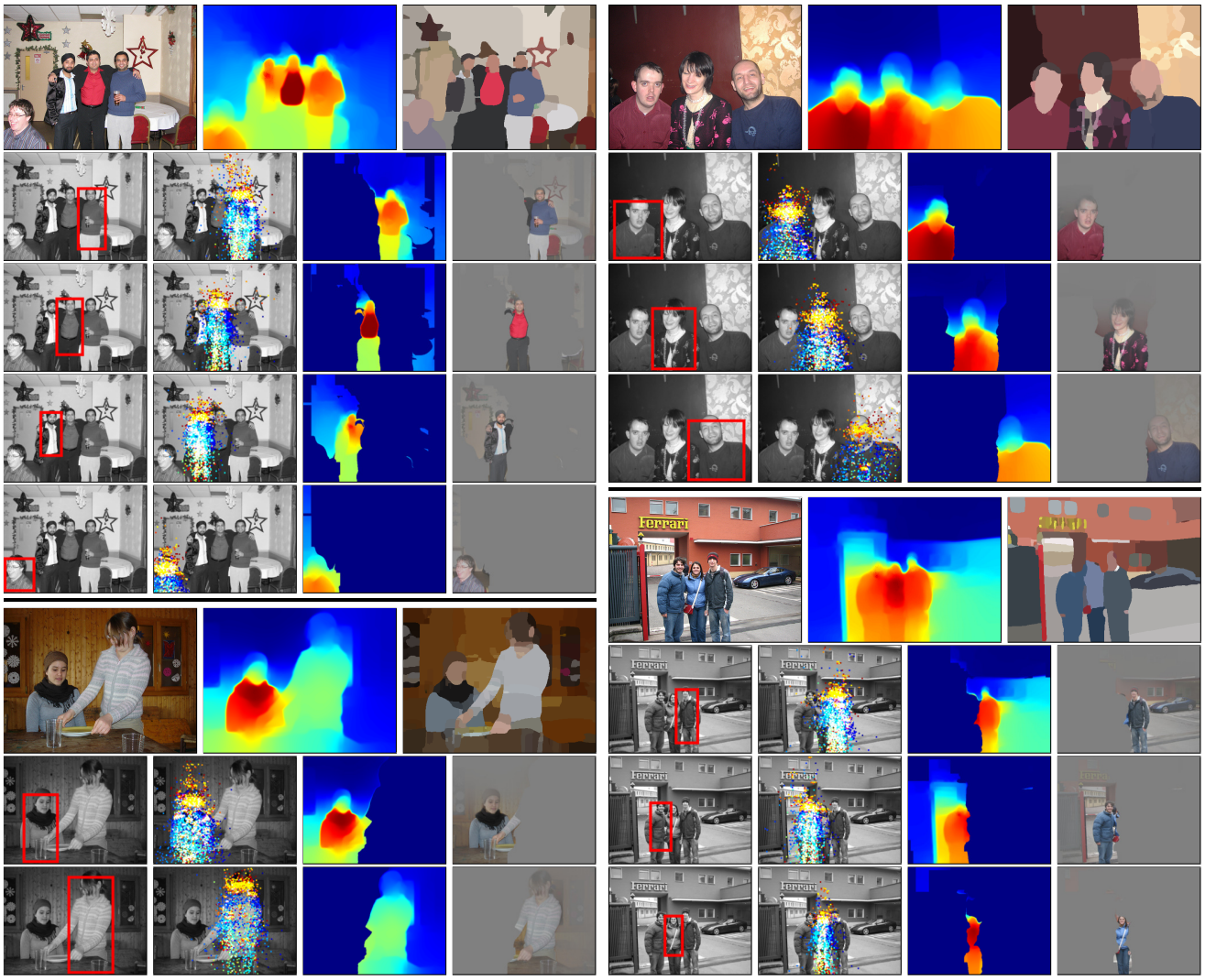

Figure 3. **Decoding eigenvectors.** *Each Subfigure - Top Row, Left:* Image. *Middle:* The angle of the leading complex eigenvector defines a global figure/ground ordering [18] as shown, with red indicating figure and blue indicating ground. Here, the notion of figure is category-specific and means belonging to the person class. *Right:* The eigenvectors define an embedding into a space in which the distance between graph nodes is perceptually meaningful. On the pixel grid, the gradient of the eigenvectors yields an edge signal [19]. From this, one can construct a hierarchical segmentation [1], one level of which is displayed here with each region taking its average color. *Bottom Rows:* Running agglomerative clustering on the eigenvector representation of the part nodes merges them into distinct object detections. Using distances between pixels and parts in the embedding space, our algorithm recovers object ownership of pixels. For each detected object instance, we show its bounding box, the body keypoint locations (head, left shoulder, right shoulder, *etc*.) predicted by the parts belonging to it (colored according to keypoint type), its instance-specific figure/ground mask, and the object extracted from the scene using this mask.

2.4.1 Figure/Ground

As previously shown [28] and applied specifically for bottom-up figure/ground organization [18], $\angle z_0$ defines a global ordering on nodes which best respects the local relative ordering relationships specified in (C, Θ). Hence, to obtain figure/ground, the transformation is simply to evaluate $\angle z_0$ on the pixel nodes. Multiplying $\angle z_0$ by Σ then translates back to the original scale in which we specified a unit separation between figure and ground.

2.4.2 Segmentation

Eigenvectors $z_1, ..., z_{m-1}$ give an embedding into $\mathbb{C}^{m-1}$ which maps globally similar nodes to similar locations. We can group nodes into discrete clusters by simply merging nearby nodes according to their distance in $\mathbb{C}^{m-1}$. An algorithm that uses this principle, while also exploiting the two-dimensional layout of the image, is to take a weighted combination of the gradients of the eigenvectors, $\nabla z_1, ..., \nabla z_{m-1}$, (evaluated on the pixel grid) in order to

recover a global contour signal [19]. Using image morphology tools, we can then construct a hierarchical segmentation from the contours [1]. The top right panel for each example in Figure 3 shows regions obtained using this process.

2.4.3 Object Detection

Just as merging pixels based on their proximity in the embedding space groups them into regions, merging parts based on proximity in this space groups them into objects. The one caveat is that not all parts belong to some object, as our part detector is imperfect. Therefore, we employ a two-step procedure of agglomeratively merging parts into potential objects and then filtering out part clusters that do not appear to be plausible object detections. Our algorithm then once again exploits the embedding representation to derive per-object instance segmentation masks.

In particular, part nodes are merged into object detections according to a weighted L_1 distance in $\mathbb{C}^{m-1}$:

$$D(q_i, q_j) = \sum_{k=1}^{m} \frac{1}{\sqrt{1-\lambda_k}} |z_k(q_i) - z_k(q_j)| \qquad (19)$$

This metric is analogous to that used for the pixel nodes, with the same eigenvalue weighting term. While merging, the distance between clusters is the maximum distance between any of their contained parts. This procedure terminates at a fixed distance threshold.

Though the partitioning of parts into distinct object hypotheses is done using only information contained in the eigenvectors, the learned object model predicts object bounding boxes and scores hypotheses based on their member parts. Hypotheses whose predicted bounding boxes overlap significantly, as measured by intersection over union, are automatically merged. We employ the linear discriminant classifier of [2] to score each hypothesis.

After accepting or rejecting hypotheses based on a score threshold, our algorithm returns to the eigenvector representation to pull out the pixels belonging to each individual object. This object-specific segmentation step is distinct from the generic region segmentation created in Section 2.4.2.

Our distance metric (19) is meaningful not only between pairs of parts but also between parts and pixels. Denote by $\{Q_i\}$ the set of confirmed object detections, where each Q_i is itself a set of parts $\{q_j\}$. We map each pixel p_k to the object containing the closest part covering it:

$$p_k \rightarrow \underset{Q_i}{\operatorname{argmin}} \left\{ \min_{\substack{q_j \in Q_i \\ p_k \in M_j}} \{D(p_k, q_j)\} \right\} \qquad (20)$$

where M_j is once again the region of the image overlapped by part q_j. This results in a partition of the figural pixels into distinct objects, as shown in the bottom third column for each example of Figure 3.

3. Experiments

We evaluate our system on the PASCAL VOC 2010 dataset and compare results to those obtained by Bourdev *et al*. [2] on the segmentation challenge for the person category. We use 150 poselet detectors. On each example, our system acts on the exact same set of part detections as this baseline top-down system, with the only difference being their coupling to low-level cues through our grouping framework. The task of each system is to assemble these parts into object hypotheses, with the baseline algorithm accomplishing this by agglomerative merging according to D_{SKL} (11) and our algorithm instead relying on distance in the embedding space (19). Both algorithms use the same hypothesis scoring function and same detection threshold.

Though our system produces pixel-level object segmentations as a byproduct of grouping, we temporarily ignore this output in order to facilitate a strict comparison. Instead, we compare object masks generated by averaging the masks of their member poselets according to the procedure given in [2]. Thus, we remove from the evaluation the factor of our object segmentations versus those obtained in the post-processing contour alignment step of [2]. We benchmark the results shown in the second column of Figure 4, even though we also generate the higher-quality segmentations visible in the later columns. This handicap ensures that any improvements on the benchmark must be due to our low-level cues assisting in part grouping and object detection.

We achieve an 11% relative boost in pixel accuracy over Bourdev *et al*. when comparing object masks for the person category on the PASCAL 2010 segmentation test set (absolute score of 39.5 compared to 35.5, each as reported by the automated PASCAL VOC evaluation server). This boost is entirely a result of our grouping framework, as neither system tested uses multi-class context or does any post-processing (though work contemporaneous with ours shows gains by exploiting such techniques [4]). Although not our focus, reclassifying each region in our output segmentation according to all available single-class cues (predicted bounding box, poselets, and figure/ground ordering) further improves our person segmentation score to 41.1.

Figures 4 and 5 show that integrating low-level information helps detection. We automatically extract detailed masks that conform precisely to object boundaries, as shown in the third column of Figure 4. Our system also handles unusual poses (top middle example of Figure 5) that cannot be segmented using the top-down poselet model alone. In addition, our system filters occluding objects, such as the bike in the first example, or the chair and desk in the last example of Figure 4, from returned person detections. Most importantly, the low-level cues assist in part grouping and allow our algorithm to find people missed by the top-down scanning of Bourdev *et al*. Figure 5 shows examples that our system pushes over the detection threshold.

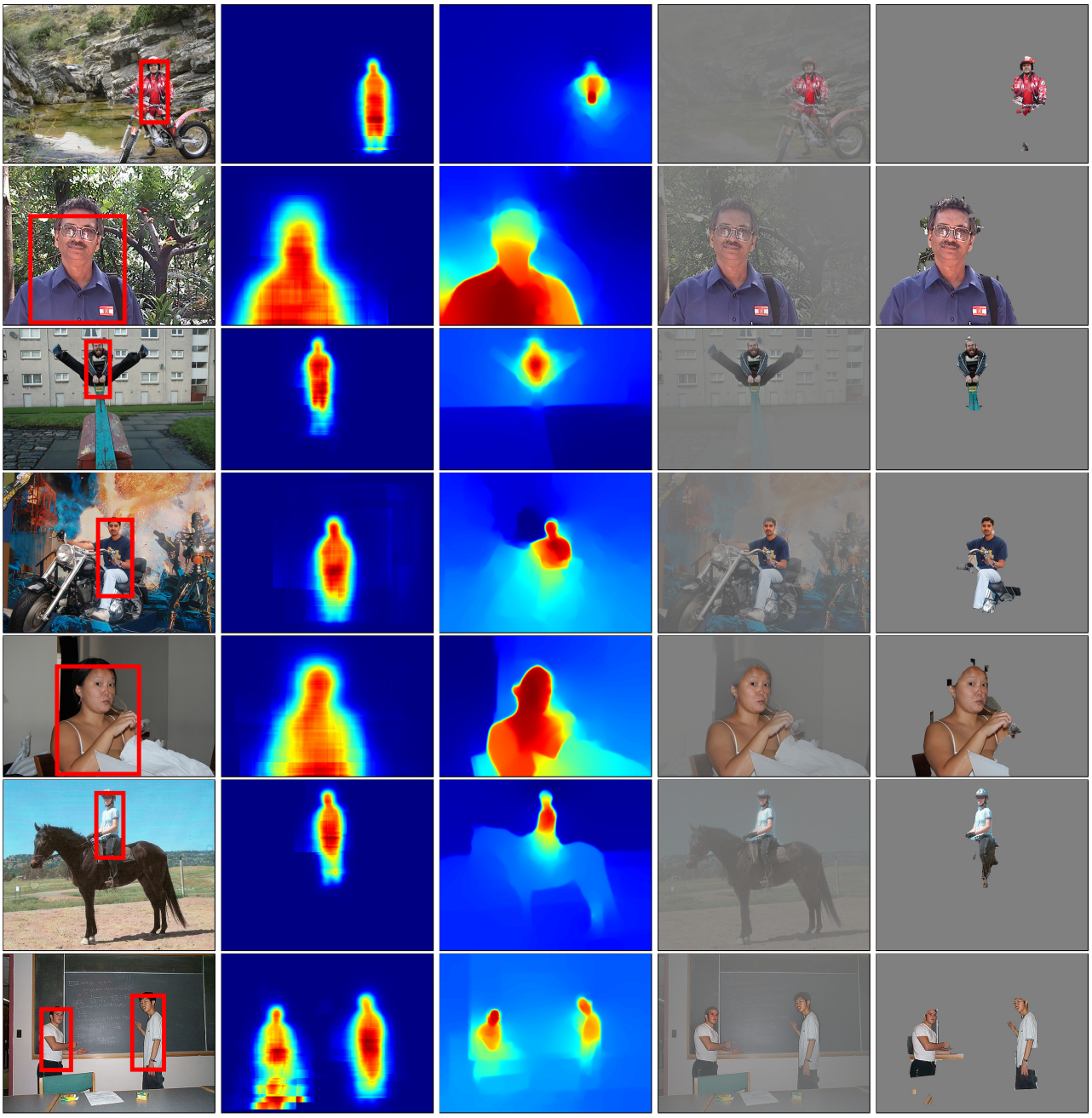

Figure 4. **Person detection results.** *Left:* Image and detection bounding boxes. *Middle Left:* Object masks formed by averaging the poselets grouped together by our globalization process. This prediction is analogous to the bottom second columns of Figure 3. *Middle:* Object masks obtained using the full representation produced by globalization. Note how these masks conform to the true object boundaries and respect occluders present in the scene. *Middle Right:* Foreground extracted by multiplying the image with object masks in the middle column. *Right:* Binary classification of regions belonging to the person category using both mask and bounding box predictions. All examples are from the PASCAL VOC 2010 test set.

4. Conclusion

Our algorithm combines segmentation cues with object detection in a soft manner. By doing so, we boost the performance of a state-of-the-art person detector. Our framework offers additional advantages not reflected in the quantitative evaluation, as it transforms an image into a representation containing a complete description of the scene in terms of figure/ground, segmentation, and object instances.

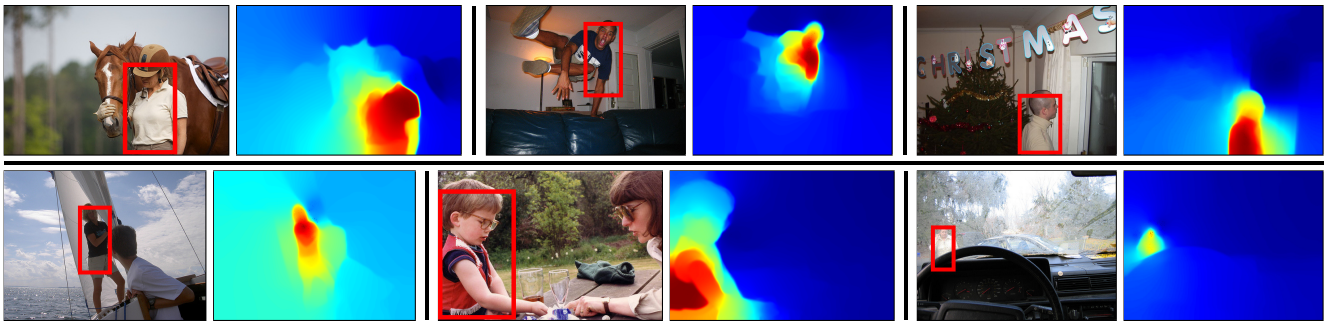

Figure 5. **Detections boosted by low-level cues.** Shown are some of the examples from the PASCAL VOC 2010 test set that are missed by the top-down poselet system but correctly detected by our joint grouping framework. Low-level cues appear to help in cases of unusual pose (top row, middle) or partial occlusion (bottom row, right). We display bounding box and figure/ground output.

Acknowledgments. ONR MURI N00014-06-1-0734, ONR MURI 1015 G NA127, and ARL Cooperative Agreement W911NF-10-2-0016 supported this work. Stella X. Yu was funded by NSF CAREER IIS-0644204 and a Clare Boothe Luce Professorship. Thanks to Jitendra Malik for suggesting poselets as a figure/ground cue and Lubomir Bourdev for providing poselet code.

References

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. *CVPR*, 2009.

[2] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. *ECCV*, 2010.

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. *ICCV*, 2009.

[4] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. *CVPR*, 2011.

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

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[7] P. Dollar, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. *BMVC*, 2010.

[8] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 2010.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009.

[10] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *CVPR*, 2003.

[11] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object recognition and localization with stable segmentations. *ECCV*, 2008.

[12] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. *ICCV*, 2009.

[13] C. Gu, J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. *CVPR*, 2009.

[14] D. Hoiem, A. A. Efros, and M. Hebert. Closing the loop on scene interpretation. *CVPR*, 2008.

[15] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. *CVPR*, 2010.

[16] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where & how many? combining object detectors and crfs. *ECCV*, 2010.

[17] J. Lim, P. Arbeláez, C. Gu, and J. Malik. Context by region ancestry. *ICCV*, 2009.

[18] M. Maire. Simultaneous segmentation and figure/ground organization using angular embedding. *ECCV*, 2010.

[19] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. *CVPR*, 2008.

[20] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. *BMVC*, 2007.

[21] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 2004.

[22] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. *ECCV*, 2010.

[23] X. Ren and J. Malik. Learning a classification model for segmentation. *ICCV*, 2003.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[25] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. *CVPR*, 2010.

[26] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. *ICCV*, 2009.

[27] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object detection for multi-class segmentation. *CVPR*, 2010.

[28] S. X. Yu. Angular embedding: from jarring intensity differences to perceived luminance. *CVPR*, 2009.

[29] S. X. Yu, R. Gross, and J. Shi. Concurrent object recognition and segmentation by graph partitioning. *NIPS*, 2002.

[30] S. X. Yu and J. Shi. Object-specific figure-ground segregation. *CVPR*, 2003.

[31] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *PAMI*, 2004.