Pattern Recognition Letters 72 (2016) 4–14

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

The three R's of computer vision: Recognition, reconstruction and reorganization☆

Jitendra Malik[a,*], Pablo Arbeláez[b], João Carreira[a], Katerina Fragkiadaki[a], Ross Girshick[c], Georgia Gkioxari[a], Saurabh Gupta[a], Bharath Hariharan[c], Abhishek Kar[a], Shubham Tulsiani[a]

[a] EECS, UC Berkeley, Berkeley, CA 94720, USA
[b] Universidad de los Andes, Bogotá 111711, Colombia
[c] Facebook, Seattle, WA 98101, USA

ARTICLE INFO

Article history:
Available online 8 February 2016

Keywords:
Object recognition
Action recognition: grouping
Segmentation
3D models
Shape reconstruction

ABSTRACT

We argue for the importance of the interaction between recognition, reconstruction and re-organization, and propose that as a unifying framework for computer vision. In this view, recognition of objects is reciprocally linked to re-organization, with bottom-up grouping processes generating candidates, which can be classified using top down knowledge, following which the segmentations can be refined again. Recognition of 3D objects could benefit from a reconstruction of 3D structure, and 3D reconstruction can benefit from object category-specific priors. We also show that reconstruction of 3D structure from video data goes hand in hand with the reorganization of the scene. We demonstrate pipelined versions of two systems, one for RGB-D images, and another for RGB images, which produce rich 3D scene interpretations in this framework.

1. Introduction

The central problems in computer vision are recognition, reconstruction and reorganization (Fig. 1).

Recognition is about attaching semantic category labels to objects and scenes as well as to events and activities. Part–whole hierarchies (partonomies) as well as category–subcategory hierarchies (taxonomies) are aspects of recognition. Fine-grained category recognition includes as an extreme case instance level identification (e.g. Barack Obama's face).

Reconstruction is traditionally about recovering three-dimensional geometry of the world from one or more of its images. We interpret the term more broadly as "inverse graphics" – estimating shape, spatial layout, reflectance and illumination – which could be used together to render the scene to produce an image.

Reorganization is our term for what is usually called "perceptual organization" in human vision; the "re" prefix makes the analogy with recognition and reconstruction more salient. In computer vision the terms grouping and segmentation are used with approximately the same general meaning.

Mathematical modeling of the fundamental problems of vision can be traced back to geometers such as Euclid [10], scientists such as Helmholtz [41], and photogrammetrists such as Kruppa [51]. In the twentieth century, the Gestaltists led by Wertheimer [85] emphasized the importance of perceptual organization. Gibson [26] pointed out the many cues which enable a moving observer to perceive the three-dimensional structure of the visual world.

The advent of computers in the middle of the twentieth century meant that one could now develop algorithms for various vision tasks and test them on images, thus creating the field of computer vision. [64] is often cited as the first paper in this field, though there was work on image processing and pattern recognition even before that. In recent years, progress has been very rapid, aided not only by fast computers, but also large annotated image collection such as ImageNet [16].

But is there a unifying framework for the field of computer vision? If one looks at the proceedings of a recent computer vision conference, one would notice a variety of applications using a wide range of techniques such as convex optimization, geometry, probabilistic graphical models, neural networks, and image processing. In the early days of computational vision, in the 1970s and 1980s, there was a broad agreement that vision could be usefully broken up into the stages of low level, mid level and high level vision. [58] is perhaps the best articulation of this point of view, with low level vision corresponding to processes such as edge detection, mid

☆ This paper has been recommended for acceptance by Rama Chellapp.

* Corresponding author. Tel.: +1 510 642 7597.
E-mail address: malik@cs.berkeley.edu, malik@eecs.berkeley.edu (J. Malik).

http://dx.doi.org/10.1016/j.patrec.2016.01.019
0167-8655/

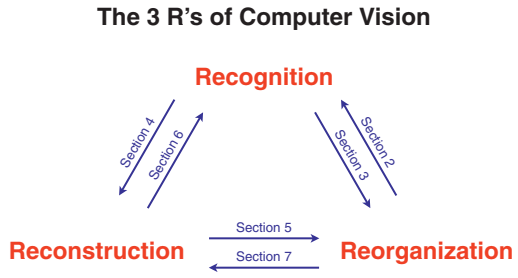

Fig. 1. The 3R's of vision: recognition, reconstruction and reorganization. Each of the six directed arcs in this figure is a useful direction of information flow.

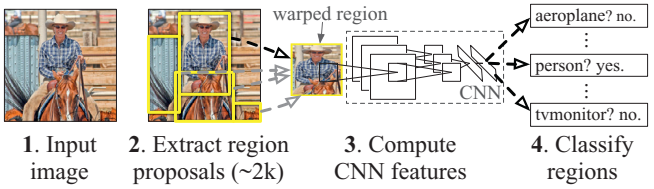

Fig. 2. R-CNN – Region-based Convolutional Network: object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional network (CNN), and then (4) classifies each region using class- specific linear SVMs. We trained an R-CNN that achieves a mean average precision (mAP) of 62.9% on PASCAL VOC 2010. For comparison, [78] report 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%. On the 200-class ILSVRC2013 detection dataset, we trained an R-CNN with a mAP of 31.4%, a large improvement over OverFeat [67], which had the previous best result at 24.3% mAP.

level vision leading to representation of surfaces, and high level vision corresponding to object recognition. The process was thought of a being primarily feed-forward and bottom up. In the 1990s, this consensus gradually dissipated. Shape-from-X modules, with the exception of those based on multiple view geometry, proved to be not robust for general images, so the bottom up construction of Marr's desired 2.5D sketch proved infeasible. On the other hand, machine learning approaches to object recognition based on sliding windows started to succeed on real world images e.g. Viola and Jones' [80] work on face detection, and these didn't quite fit Marr's paradigm.

Back in the 1990s, one of us, [57] argued that grouping and recognition ought to be considered together. Bottom up grouping could produce candidates for consideration by a recognition module. Another paper from our group, [63] advocated the use of superpixels for a variety of tasks. This got some traction e.g. multiple segmentation hypotheses were used by Hoiem et al. [42] to estimate the rough geometric scene structure and by Russell et al. [66] to automatically discover object classes in a set of images, and Gu et al. [32] showed what were then state of the art results on the ETH-Z dataset. But the dominant paradigm remained that of sliding windows, and the state of the art algorithms on the PASCAL VOC challenge through 2011 were in that paradigm.

This has changed. The "selective search" algorithm of [78] popularized the multiple segmentation approach for object detection by showing strong results on PASCAL object detection. EdgeBoxes [88] outputs high-quality rectangular (box) proposals quickly (~0.3 s per image). Other methods focus on pixel-wise segmentation, producing regions instead of boxes. Top performing approaches include CPMC [12], RIGOR [45], MCG [4], and GOP [49]. For a more in-depth survey of proposal algorithms, [43] provide an insightful meta-evaluation of recent methods.

In this paper we propose to go much further than the link between recognition and reorganization. That could be done with purely 2D reasoning, but surely our final percept must incorporate the 3D nature of the world? We will highlight a point of view that one of us (Malik) has been advocating for several years now, that instead of the classical separation of vision into low level, mid level and high level vision, it is more fruitful to think of vision as resulting from the interaction of three processes: recognition, reconstruction and reorganization which operate in tandem, and where each provides input to the others and fruitfully exploits their output. We aim for a grand unified theory of these processes, but in the immediate future it may be best to model various pairwise interactions, giving us insight into the representations that prove most productive and useful. In the next six sections, we present case studies which make this point, and we conclude with a pipeline which puts the different stages together.

Note that the emphasis of this paper is on the relationship between the 3R's of vision, which is somewhat independent of the (very important) choice of features needed to implement particular algorithms. During the 1970s and 1980s, the practice of computer vision was dominated by features such as edges and corners which offered the benefit of massive data compression, a necessity in a time when computing power was orders of magnitude less than today. The community moved on to the use of linear filters such as Gaussian derivatives, Gabor and Haar wavelets in the 1990s. The next big change was the widespread use of histogram based features such as SIFT [55] and HOG [15]. While these dominated for more than a decade, we are now completing yet another transition, that to "emergent" features from the top layers of a multilayer convolutional neural network [53] trained in a supervised fashion on a large image classification task. Neural networks have proved very compatible to the synthesis of recognition, reconstruction and reorganization.

2. Reorganization helps recognition

As noted earlier, the dominant approach to object detection has been based on sliding-window detectors. This approach goes back (at least) to early face detectors [79], and continued with HOG-based pedestrian detection [15], and part-based generic object detection [20]. Straightforward application requires all objects to share a common aspect ratio. The aspect ratio problem can be addressed with mixture models (e.g. [20]), where each component specializes in a narrow band of aspect ratios, or with bounding-box regression (e.g. [20,67]).

The alternative is to first compute a pool of (likely overlapping) image regions, each one serving as a candidate object, and then to filter these candidates in a way that aims to retain only the true objects. By combining this idea with the use of convolutional network features, pretrained on an auxiliary task of classifying ImageNet, we get the Region-based Convolutional Network (R-CNN) which we describe next.

At test time, R-CNN generates around 2000 category-independent region proposals for the input image, extracts a fixed-length feature vector from each proposal using a convolutional neural network (CNN) [53], and then classifies each region with category-specific linear SVMs. We use a simple warping technique (anisotropic image scaling) to compute a fixed-size CNN input from each region proposal, regardless of the region's shape. Fig. 2 shows an overview of a Region-based Convolutional Network (R-CNN) and Table 1 presents some of our results.

R-CNNs scale very well with the number of object classes to detect because nearly all computation is shared between all object categories. The only class-specific computations are a reasonably small matrix-vector product and greedy non-maximum suppression. Although these computations scale linearly with the number of categories, the scale factor is small. Measured empirically, it takes only 30 ms longer to detect 200 classes than 20 classes on a CPU, without any approximations. This makes it feasible to

Table 1
Detection average precision (%) on VOC 2010 test. AlexNet is the CNN architecture from [50] and VGG16 from [71]. R-CNNs are most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression (BB) is described in [29]. DPM and SegDPM use context rescoring not used by the other methods. SegDPM and all R-CNNs use additional training data.

VOC 2010 test	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV	mAP
DPM v5 [30]	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [78]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [83]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [21]	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN AlexNet	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN AlexNet BB	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	53.7
R-CNN VGG16	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2	59.8
R-CNN VGG16 BB	**79.3**	**72.4**	**63.1**	**44.0**	**44.4**	**64.6**	**66.3**	**84.9**	**38.8**	**67.3**	**48.4**	**82.3**	**75.0**	**76.7**	**65.7**	**35.8**	**66.2**	**54.8**	**69.1**	**58.8**	**62.9**

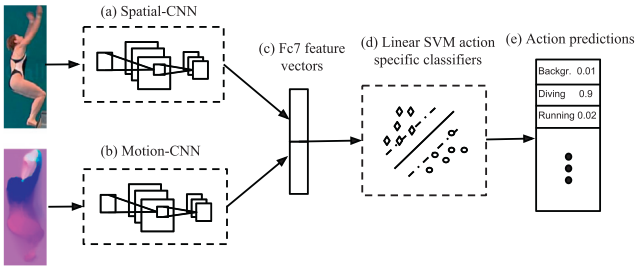

Fig. 3. Finding Action Tubes: our basic model for action detection. We use action specific SVM classifiers on spatio-temporal features. The features are extracted from the fc7 layer of two CNNs, spatial-CNN and motion-CNN, which were trained to detect actions using static and motion cues, respectively.

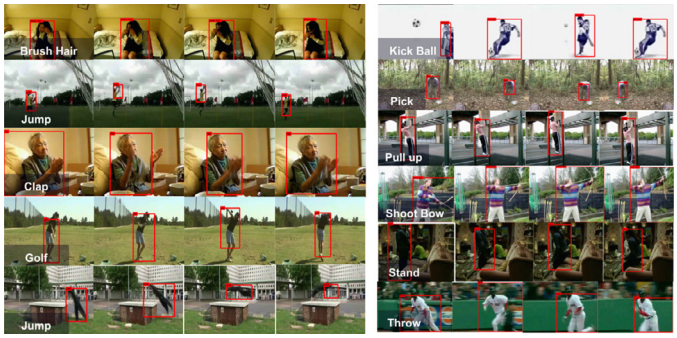

Fig. 4. Results of Action Tubes on J-HMDB. We show the regions with red. The predicted action label is overlaid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

rapidly detect tens of thousands of object categories without any modifications to the core algorithm.

Despite this graceful scaling behavior, an R-CNN can take 10–45 s per image on a GPU, depending on the network used, since each region is passed through the network independently. Recent work from [40] ("SPPnet") improves R-CNN efficiency by sharing computation through a feature pyramid, allowing for detection at a few frames per second. Building on SPPnet, [27] shows that it is possible to further reduce training and testing times, while improving detection accuracy and simplifying the training process, using an approach called "Fast R-CNN." Fast R-CNN reduces testing times to 50–300 ms per image, depending on network architecture.

Turning to the task of action recognition, R-CNN can be adapted to detect actions in space and time efficiently with videos considered as input, as we show in [31].

Most work in the field [1,62,84] has focused on the task of *action classification*, i.e. "Is an action present in the video?" The state-of-the-art approach in this field by [81] uses dense point trajectories, where features are extracted from regions tracked using optical flow. More recently, [70] use two-stream CNNs to classify videos. The first CNN operates on the RGB frames, while the second CNN on the corresponding optical flow. They weight the predictions from the two CNNs and average the scores across frames to make the final prediction for the whole video. On the other hand, the task of *action detection*, i.e. "Is there an action and where is it in the video?" is less studied despite its higher potential for practical applications.

Inspired by the recent progress in object detection, we build models which localize and classify actions in video in both space and time. We call our predictions *Action Tubes*. Fig. 3 shows the design of our action models. We start from bottom-up region proposals, similar to R-CNN. We train two CNNs, one on the RGB signal of the region (spatial-CNN) and one on the corresponding optical flow signal (motion-CNN). We combine the fc7 features of the two networks to train SVM classifiers to discriminate among the actions and the background. To produce temporarily coherent predictions, *Action Tubes*, we link action detections in time.

As mentioned earlier, R-CNN can be quite slow since each region is processed independently. One way around it is to share computation between regions. In the case of video, the motion signal is a strong cue and can be used to extract salient regions. Indeed if regions which don't contain enough motion are discarded, 85% of the computation can be saved with a loss of only 4% in recall.

Our detection pipeline makes predictions at every frame. Our goal is to align those predictions in time to produce detection tubes which follow the action across frames. For linking we use the spatial overlap of predictions in consecutive frames as well as their confidence scores. We formulate the problem using dynamic programming and infer the regions which comprise the Action Tubes. Fig. 4 shows examples of Action Tubes on J-HMDB [47].

Quantitatively, Table 2 shows the performance of Action Tubes on J-HMDB. The combination of the spatial- and motion-CNN performs significantly better compared to the single CNNs, showing the significance of static and motion cues for the task of action recognition. On UCF Sports [65], we compare Action Tubes with other approaches. Fig. 5 shows AUC for various values of intersection-over-union, which define the threshold of spatial overlap with the ground truth for a correct prediction.

Finally, we demonstrate that Action Tubes improve action classification compared to the holistic approach, where spatial-CNN is trained on the RGB frames and motion-CNN on the optical flow, similar to [70]. On J-HMDB, the holistic approach achieves an accuracy of 56.5% while Action Tubes yield 62.5%. This shows that focusing on the actor is of great importance both for the case of action detection and action classification.

3. Recognition helps reorganization

Object proposal generation methods such as those described earlier typically rely on the coherence of color and texture to segment the image out into likely object candidates. However, such

Table 2
Results and ablation study on J-HMDB (averaged over the three splits). We report *AP* for the spatial and motion component and their combination (full). The combination of the spatial- and motion-CNN performs significantly better, showing the significance of static and motion cues for the task of action recognition.

AP (%)	Brush hair	Catch	Clap	Climb stairs	Golf	Jump	Kick ball	Pick	pour	Pullup	push	Run	Shoot ball	Shoot bow	Shoot gun	Sit	Stand	Swing baseball	Throw	Walk	Wave	Mean
spatial-CNN	67.1	**34.4**	37.2	36.3	93.8	7.3	14.4	29.6	80.2	93.9	17.4	10.0	8.8	71.2	**45.8**	17.7	11.6	38.5	20.4	40.5	19.4	37.9
motion-CNN	66.3	16.0	**60.0**	51.6	88.6	**18.9**	10.8	23.9	83.4	96.7	18.2	17.2	**14.0**	84.4	19.3	**72.6**	**61.8**	**76.8**	17.3	46.7	14.3	45.7
full	**79.1**	33.4	53.9	**60.3**	**99.3**	18.4	**26.2**	**42.0**	**92.8**	**98.1**	**29.6**	**24.6**	13.7	**92.9**	42.3	67.2	57.6	66.5	**27.9**	**58.9**	**35.8**	**53.3**

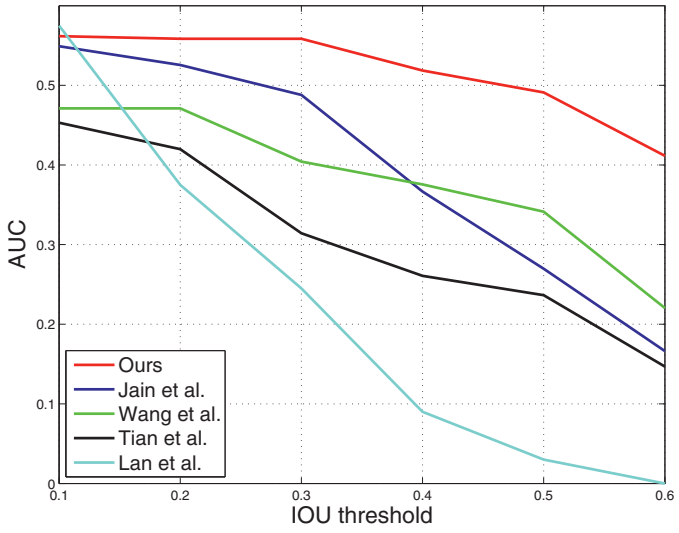

Fig. 5. AUC on UCF Sports for various values of intersection-over-union threshold of σ (x-axis). Red shows our approach. We consistently outperform other approaches in [46], [82] [72] and [52]. The biggest improvement is being achieved at high values of overlap ($\sigma \geq 0.4$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. An image of a dog (left), bottom-up contours (center) and segmentation predicted using top-down information (right).

cues can often make mistakes. For instance, the boundary between the dog in Fig. 6 and the wall is barely perceptible, while the meaningless contour between the dog's face and its torso is sharp. However, once an object detection approach such as R-CNN detects the dog, we can bring to bear our knowledge about dogs and segment out the dog much more accurately. Recognition of the dog should thus aid us in accurate segmentation. See [6] for an early implementation of this idea.

We explored this relationship between recognition and reorganization in two papers [37,38] where we presented the Simultaneous Detection and Segmentation (SDS) task: detect all instances of a category in an image and, for each instance, correctly mark the pixels that belong to it. Our algorithm for this task consists of the following steps:

1. We use R-CNN (Section 2) to first detect objects of all categories along with coarse bounding boxes. We keep the non-maximum suppression threshold low to avoid prematurely removing any correct detection.
2. For each detected object, we make a *top–down, category-specific* prediction of the figure-ground.
3. We then use the predicted segments to *rescore* the detections.
4. We finally run a round of non-max suppression to remove duplicate detections.

Next we describe how we predict the figure-ground mask for each detection and how we use the predicted segments to rescore the detections.

3.1. Top-down prediction of figure-ground

Given a detection, we predict a heatmap on the bounding box of the detection, which we splat onto the image. This heatmap encodes the probability that a particular location is inside the object. Thus, the task of predicting figure-ground masks reduces to classifying each location as being inside the object or not.

As in R-CNN, we use features from the CNN to do this classification. Typically, recognition algorithms such as R-CNN use the output of the last layer of the CNN. This makes sense when the task is assigning category labels to images or bounding boxes: the last layer is the most sensitive to category-level semantic information and the most invariant to "nuisance" variables such as pose, illumination, articulation, precise location and so on. However, when the task we are interested in is finer-grained, as is the case in SDS, these nuisance variables are precisely what carries the signal. For such applications, the top layer is thus *not* the optimal representation.

The information that is generalized over in the top layer is present in intermediate layers, but intermediate layers are also much less sensitive to semantics. For instance, bar detectors in early layers might localize bars precisely but cannot discriminate between bars that are horse legs and bars that are tree trunks. This observation suggests that reasoning at multiple levels of abstraction and scale is necessary, mirroring other problems in computer vision (such as optical flow) where reasoning across multiple levels has proved beneficial.

To capture such reasoning, we developed the hypercolumn representation. We define the "hypercolumn" at a given input location as the outputs of all CNN units that lie above that location at all layers of the CNN, stacked into one vector. (Because adjacent layers are correlated, in practice we need not consider all the layers but can simply sample a few.) Fig. 7 shows a visualization of the idea. We borrow the term "hypercolumn" from neuroscience, where it is used to describe a set of V1 neurons sensitive to edges at multiple orientations and multiple frequencies arranged in a columnar structure [44]. The notion of combining predictions from multiple CNN layers is also described in [54,56,68]. There has also been prior work on semantic segmentation using CNNs [19], though such work does not group pixels into individual instances.

Given such a hypercolumn representation for each pixel, we can then use these as features to train pixel classifiers to label each pixel as figure-vs-ground. The full details of the implementation are in [38].

3.2. Scoring region candidates

Once we have produced a segmentation for each detection, we now rescore the detection. As for action recognition, we use a

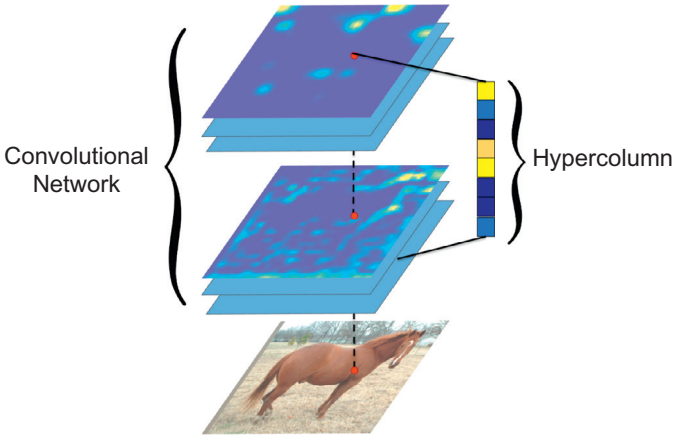

Fig. 7. The hypercolumn representation. The bottom image is the input, while the other images represent feature maps of different layers in the CNN. The hypercolumn at a pixel is the vector of activations of all units that lie above that pixel.

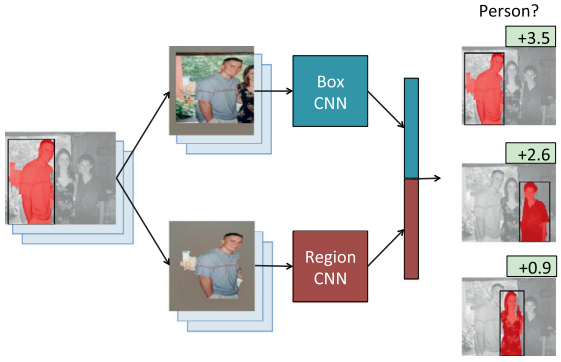

Fig. 8. The region classification network.

Table 3
Performance of various systems on the SDS task.

	Network	AP^r at 0.5	AP^r at 0.7
Top layer only	AlexNet	44.0	16.3
Hypercolumn (Layers 7, 4, 2)	AlexNet	49.1	29.1
Hypercolumn (Layers 7, 4, 3)	VGG16	56.5	37.0
+Rescore	VGG16	60.0	40.4

two-stream architecture as shown in Fig. 8. The first stream operates on the cropped bounding box of the detection (the "box" pathway) while the second stream operates on the cropped bounding box with the predicted background masked (the "region" pathway). Features from the two streams are concatenated and passed into the classifier, and the entire network is trained jointly end-to-end.

3.3. Evaluation metric

We evaluate our performance using an extension of the bounding box detection metric [37,73,87]. The algorithm produces a ranked list of detections where each detection comes with a predicted segmentation. A detection is correct if its *segmentation* overlaps with the *segmentation* of a ground truth instance by more than a threshold. As in the classical bounding box task, we penalize duplicates. With this labeling, we compute a precision recall (PR) curve, and the average precision (AP), which is the area under the curve. We call the AP computed in this way AP^r. We report AP^r at overlap thresholds of 0.5 and 0.7 in Table 3.

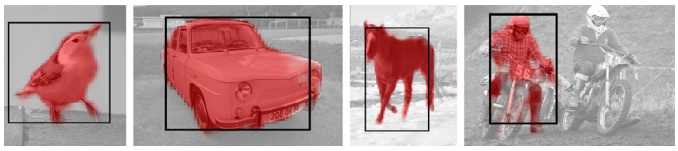

Fig. 9. Qualitative results. In black is the bounding box and in red is the predicted segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

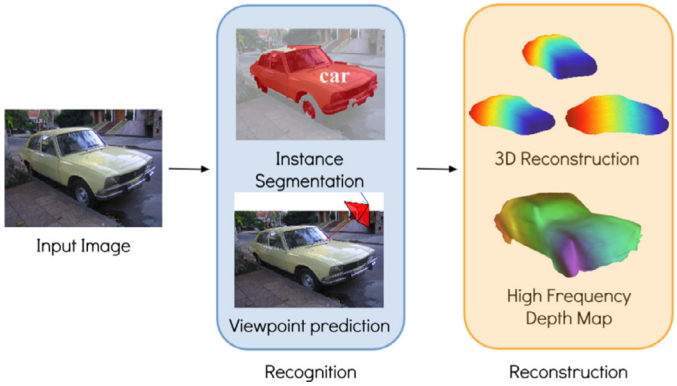

Fig. 10. Automatic object reconstruction from a single image. Our method leverages estimated instance segmentations and predicted viewpoints to generate a full 3D mesh and high frequency 2.5D depth maps.

We observe that

1. Using three different layers to make the figure-ground prediction almost doubles AP^r at 0.7, validating the intuition behind the hypercolumn representation.
2. Using a larger network leads to large performance gains (about 8 points).
3. Rescoring detections using segment-based features also provides a significant boost (about 3 points).

Qualitative segmentations are shown in Fig. 9.

4. Recognition helps reconstruction

Consider Fig. 10. As humans, we can easily perceive the 3D shape of the shown object, even though we might never have seen this particular object instance. We can do this because we don't experience this image *tabula rasa*, but in the context of our "remembrance of things past". We can recognize that this is the image of a car as well as estimate the car's 3D pose. Previously seen cars enable us to develop a notion of the 3D shape of cars, which we can project to this particular instance using its determined pose. We also specialize our representation to this particular instance (e.g. any custom decorations it might have), signaling that both top-down and bottom-up cues influence our percept – see also [59].

In order to operationalize these observations into an approach that can perceive the shape of novel instances we need recognition systems that can infer the location and pose of objects as well as category-specific shape models that capture knowledge about previously seen objects. We previously described a simultaneous detection and segmentation (SDS) system that can give us the instance segmentations of the objects present in the image. We now describe our pose (viewpoint) estimation, originally published in [77], and category-specific shape modeling systems – from [48]. We then show that building upon these object recognition and reconstruction systems, 3D object reconstruction from a single image in the wild can be readily addressed as shown in Fig. 10.

Fig. 11. Viewpoint predictions for unoccluded groundtruth instances using our algorithm. The columns show 15th, 30th, 45th, 60th, 75th and 90th percentile instances respectively in terms of the error. We visualize the predictions by rendering a 3D model using our predicted viewpoint.

Table 4
Mean performance of our approach for simultaneous detection and viewpoint estimation.

	AVP			
Number of bins	**4**	**8**	**16**	**24**
[86]	19.5	18.7	15.6	12.1
[60]	23.8	21.5	17.3	13.6
[25]	24.1	22.3	17.3	13.7
Ours	**49.1**	**44.5**	**36.0**	**31.1**

4.1. Viewpoint prediction

We formulate the problem of viewpoint prediction as predicting three Euler angles (azimuth, elevation and cyclo-rotation) corresponding to the instance. We train a CNN based architecture which can implicitly capture and aggregate local evidences for predicting the Euler angles to obtain a viewpoint estimate. Our CNN architecture additionally enables learning a shared feature representation across all categories.

We visualize the predictions of our approach in Fig. 11. It can be seen that our predictions are accurate even for the 75th percentile instances. To evaluate for the task of simultaneous detection and viewpoint estimation, we use the AVP metric introduced by Xiang et al. [86]. Each detection candidate has an associated viewpoint and the detection is labeled correct if it has a correct predicted viewpoint bin as well as a correct localization (bounding box IoU >0.5). As shown in Table 4, our approach significantly outperforms previous methods.

4.2. Category-specific shape models

Unlike prior work that assumes high quality detailed annotations [13], we are interested in reconstructing objects "in the wild". At the core of our approach are deformable 3D models that can be learned from 2D annotations available in existing object detection datasets and can be driven by noisy automatic object segmentations. These allow us to overcome the two main challenges to object reconstruction in the wild: (1) Detection datasets typically have many classes and each may encompass wildly different shapes, making 3D model acquisition expensive. (2) 3D shape inference should be robust to any small imperfections that detectors produce, yet be expressive enough to represent rich shapes. We use the learned deformable 3D models to infer the shape for each detection in a novel input image and further complement it with a bottom-up module for recovering high-frequency shape details. The learning and inference steps are described below.

Fig. 12 illustrates our pipeline for learning shape models from just 2D training images, aided by ground truth segmentations and a few keypoints. We first use the NRSfM framework of [76], extended to incorporate silhouette information, to jointly estimate

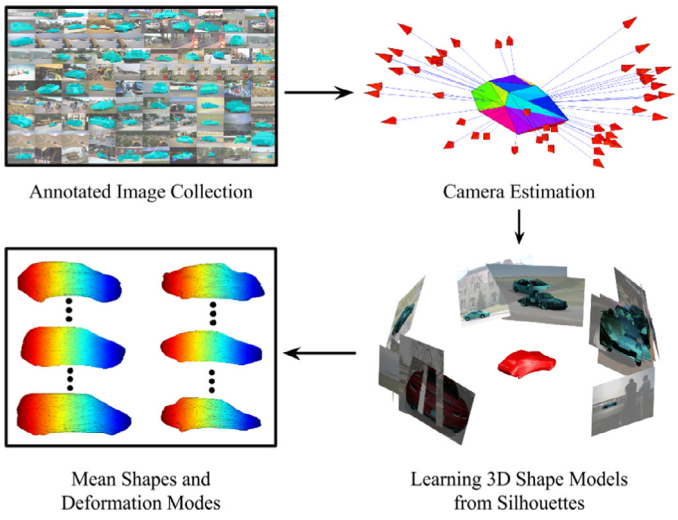

Fig. 12. Overview of our training pipeline for learning deformable shape models.

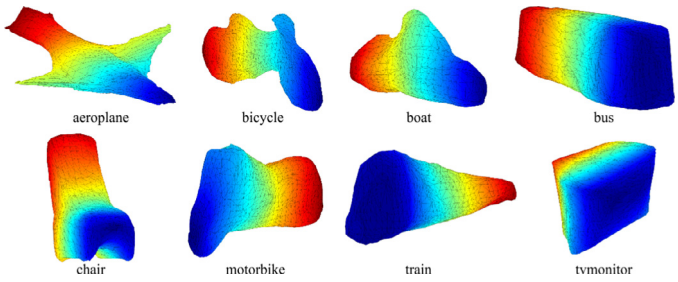

Fig. 13. Mean shapes learnt for rigid classes in PASCAL VOC using our formulation. Color encodes depth when viewed frontally.

the camera viewpoints (rotation, translation and scale) for all training instances in each class.

We then learn the category level shape models by optimizing over a deformation basis of representative 3D shapes that best explain all silhouettes, conditioned on the camera viewpoints. We model our category shapes as deformable point clouds – one for each subcategory of the class. Our shape model $M = (\bar{S}, V)$ comprises of a mean shape $\bar{S}$ (learnt mean shapes for several classes are shown in Fig. 13) and linear deformation bases $V = \{V_1, \ldots, V_K\}$. We formulate our energy primarily based on image silhouettes and priors on natural shapes. These energies enforce that the shape for an instance is consistent with its silhouette (E_s, E_c), shapes are locally consistent (E_l), normals vary smoothly (E_n) and the deformation parameters are small ($\|\alpha_{ik} V_k\|_F^2$). Finally, we solve the constrained optimization in Eq. (1) using block-coordinate descent to learn the category level shape model. We refer the reader to [48] for detailed descriptions of the energy

Fig. 14. Contour detection: color image, depth image, contours from color image, contours from colour and depth image, contour labels. (For interpretation of the references to color in this figure text, the reader is referred to the web version of this article.)

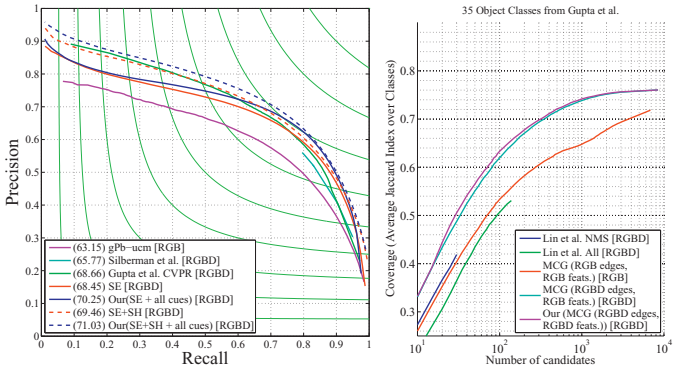

Fig. 15. Additional depth information from a Kinect like sensor improves the quality of bottom-up contours and region proposals.

formulation and optimization.

$$\min_{\bar{S},V,\alpha} \quad E_l(\bar{S},V) + \sum_i \left(E_s^i + E_{kp}^i + E_c^i + E_n^i + \sum_k (\|\alpha_{ik} V_k\|_F^2) \right) \quad (1)$$

$$\text{subject to: } S^i = \bar{S} + \sum_k \alpha_{ik} V_k$$

5. Reconstruction helps reorganization

RGB-D sensors like the Microsoft Kinect, provide a depth image in addition to the RGB image. We use this additional depth information as 'reconstruction' input and study the reorganization problems. In particular we study the problem of contour detection and region proposal generation.

5.1. Contour detection

Depth images have C^0 and C^1 discontinuities corresponding to depth and surface normal edges in the scene. We design local gradient functions to capture both these sets of discontinuities. We call these gradients Depth Gradients (*DG*) and Normal Gradients *NG*. We distinguish between convex and concave normal gradients NG_+ and NG_-. This distinction is useful given concave normal gradient edges (such as those between the bed and the floor in Fig. 14) are more likely to correspond to object boundaries than convex gradient edges (such as those within the bed). We compute these gradients by estimating local planar approximations and computing the distance between these planes to compute *DG* and the angle between them to estimate NG_+ and NG_-.

We can then use this set of estimated gradients from depth images in existing contour detection systems for RGB and RGB-D images. We experimented with two such state-of-the-art systems, RGB only gPb-ucm [3] and RGB-D Structured Edges [17] on the NYUD2 dataset [69], and see improvements over both these methods. We show the precision recall curves for contour detection in Fig. 15(left) and see that adding our gradients improves performance across the board [34,36].

Fig. 14 shows visualizations for the obtained contours. As expected, we get less distracted with albedo, recall faint but important scene surface boundaries, weigh important object boundaries more, and get more complete objects out. We can also go a step further and label each boundary as being an occlusion boundary, convex normal edge (blue) or a concave normal edge (green).

5.2. Region proposals

With this improved contour signal, we can generate region proposals for RGB-D images. For this we generalize MCG [4] which is the state-of-the-art algorithm for generating region proposals on RGB images, to RGB-D images. MCG works by (a) completing contours to form closed regions at multiple scales, (b) combinatorially combining these closed regions to produce millions of candidates, and (c) re-ranking these candidates to obtain a ranked list of about 2000 regions.

We generalize MCG in the following ways: (a) we use our improved RGB-D contour signal from Section 5.1 and (b) use features computed from the depth image for re-ranking the candidates. We measure the quality of our region proposals using standard region overlap metrics in Fig. 15. We plot the number of regions on *X*-axis and the average best overlap on the *Y*-axis, and observe that with only 200 RGB-D region proposals per image, we can obtain the same average best overlap as one would get with 2000 regions using RGB based MCG [36], illustrating the quality of our contours and the utility of using reconstruction input in the form of depth images for the re-organization problem.

6. Reconstruction helps recognition

We can also use the depth image from a Kinect sensor to aid recognition. More specifically, we study how reconstruction input in the form of a depth image from a RGB-D sensor can be used to aid performance for the task of object detection.

We conduct these experiments on the NYUD2 dataset [69]. The set of categories that we study here included indoor furniture categories like chairs, beds, sofas, and tables.

We frame this problem as a feature learning problem and use a convolutional neural network to learn features from RGB-D images. To do this, we build on the R-CNN work [28]. We propose a novel geocentric embedding of depth images into horizontal disparity, height above ground and angle with gravity (denoted HHA) (we automatically estimate the direction of gravity [34] and the height above ground from the depth image). This embedding captures basic physical properties that defines such object categories. For instance, consider a chair, it is characterized by a horizontal surface of a particular area at a particular height and a vertical surface of a particular area in a specific location relative to the horizontal surface. Our proposed embedding exposes this information to the feature learning algorithm.

In our experiments (reported in [36]), we observe the following: (a) a CNN learned on color images (in our case a CNN [50] learned for image classification on ImageNet [16]) learns features

Table 5
Results for object detection on NYUD2 [69]: we report detection average precision, and compare against three baselines: RGB DPMs, RGBD-DPMs, and RGB R-CNN. For details refer to [36].

	Mean	Bathtub	Bed	Book shelf	Box	Chair	Counter	Desk	Door	Dresser	Garbage bin	Lamp	Monitor	Night stand	Pillow	Sink	Sofa	Table	Television	Toilet
RGB DPM	**9.0**	0.9	27.6	9.0	0.1	7.8	7.3	0.7	2.5	1.4	6.6	22.2	10.0	9.2	4.3	5.9	9.4	5.5	5.8	34.4
RGBD-DPM	**23.9**	19.3	56.0	17.5	0.6	23.5	24.0	6.2	9.5	16.4	26.7	26.7	34.9	32.6	20.7	22.8	34.2	17.2	19.5	45.1
RGB R-CNN	**22.5**	16.9	45.3	28.5	0.7	25.9	30.4	9.7	16.3	18.9	15.7	27.9	32.5	17.0	11.1	16.6	29.4	12.7	27.4	44.1
Our	**37.3**	**44.4**	**71.0**	**32.9**	**1.4**	**43.3**	**44.0**	**15.1**	**24.5**	**30.4**	**39.4**	**36.5**	**52.6**	**40.0**	**34.8**	**36.1**	**53.9**	**24.4**	**37.5**	**46.8**

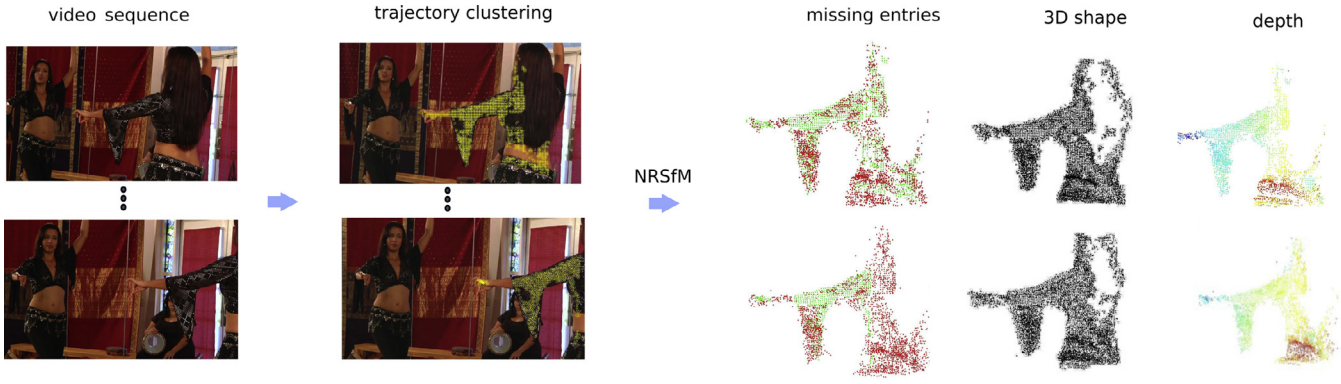

Fig. 16. Segmentation-based non-rigid structure from motion. Occluded trajectories on the belly dancer, that reside beyond the image border, are completed. (For interpretation of the references to color in this figure text, the reader is referred to the web version of this article.)

which are general enough and transfer well to depth images, and (b) using our proposed HHA embedding boosts performance over just using the depth image. Finally, when we combine our learned features on depth images with those coming from color images, we achieve state-of-the-art performance for object detection [36] (Table 5). To summarize the results briefly, RGB based Deformable Part Models (DPM) [20] (long standing state-of-the-art detection method) give a mean average precision of 9.0%, RGB-D DPMs [33] boost performance to 23.9%, current state-of-the-art RGB detection method R-CNN gives 22.5%. In contrast, our method which effectively learns features from depth images, achieves a mean average precision of 37.3%. The performance for the challenging chair category is at 43.3%, up from 7.8% a couple of years ago, taking perception a step closer to being useful for robotics.

7. Reorganization helps reconstruction

Consider Fig. 16. The leftmost column depicts a video scene that contains multiple moving objects. We want to reconstruct their spatio-temporal 3D shapes. Extensive literature exists on reconstructing static scenes from monocular uncalibrated videos, a task also known as rigid Structure-from-Motion (SfM). Some works employ scaled orthographic cameras with rank shape priors, such as the seminal factorization work of [75], while others assume perspective cameras and make use of epipolar constraints [61]. The case of deforming objects (a.k.a. Non-Rigid Structure-from-Motion (NRSfM)) has attracted less attention. Most NRSfM approaches assume as input pre-segmented, full-length point trajectories extracted from the deforming surface [2,7,14,24,39].

In our work [22], we jointly study video segmentation and 3D reconstruction of deforming objects and present a multiple hypotheses approach that deals with inaccurate video segments and temporally incomplete point tracks. For each object, our input is a trajectory matrix that holds x, y pixel coordinates of the object's 2D point trajectories. Our output is a 3D trajectory matrix with the corresponding X, Y, Z point coordinates, and a camera pose matrix that holds the per frame camera rotations. We assume scaled orthographic cameras which means that camera rotations correspond to truncated rotation matrices. Extracting 3D shape from monocular input is an under-constrained problem since many combinations of 3D point clouds and camera viewpoints give rise to the same 2D image projections. We use low-rank 3D shape priors to fight this ambiguity, similar to previous works [7].

An overview of our approach is presented in Fig. 16. Given a video sequence, we compute dense optical flow point trajectories and cluster them using normalized cuts on 2D motion similarities. We compute multiple trajectory clusterings to deal with segmentation ambiguities. In each cluster, 2D trajectories will be temporally incomplete, Fig. 16 3rd column shows in green the present and in red the missing trajectory entries for the belly dancer trajectory cluster. For each trajectory cluster, we first complete the 2D trajectory matrix using low-rank matrix completion, as in [9,11]. We then recover the camera poses through a rank 3 factorization and Euclidean upgrade of the camera pose matrix, as in [75]. Last, keeping the camera poses fixed, we minimize the reprojection error of the observed trajectory entries along with the nuclear norm of the 3D shape, similar to [14], using the nuclear norm regularized least squares algorithm of [74].

Reconstruction needs to be robust to segmentation mistakes. Motion trajectory clusters are inevitably polluted with "bleeding" trajectories that, although they reside on the background, anchor on occluding contours. We use morphological operations to discard such trajectories that do not belong to the shape subspace and confuse reconstruction.

A byproduct of our NRSfM algorithm is trajectory completion. The recovered 3D time-varying shape is backprojected in the image and the resulting 2D trajectories are completed through deformations, occlusions or other tracking ambiguities, such as lack of texture.

Fig. 17 presents reconstruction results of our approach in videos from two popular video segmentation benchmarks, VSB100 [23] and Moseg [8], that contain videos from Hollywood movies and Youtube. For each example we show (a) the trajectory cluster, (b) the present and missing trajectory points, and (c) the depths of the visible (as estimated from ray casting) points, where red and blue denote close and far respectively. Our work was the first to show dense non-rigid reconstructions of objects from real videos, without employing object-specific shape priors.

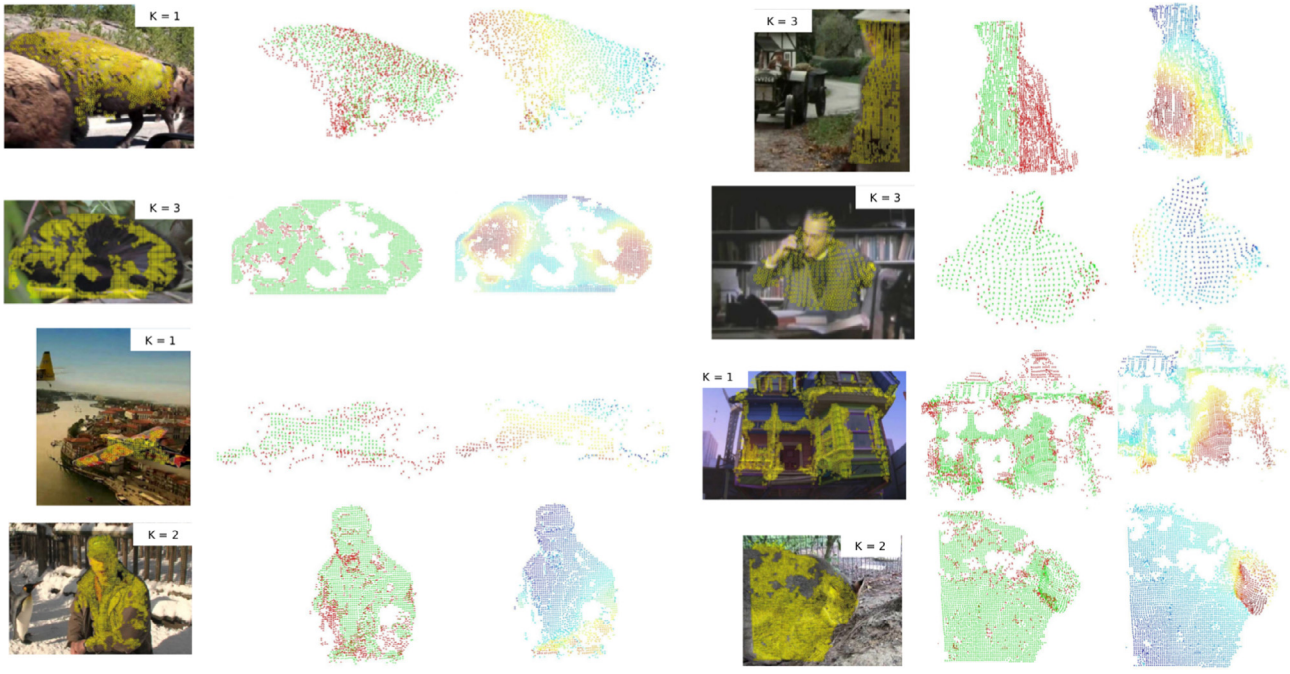

Fig. 17. Reconstruction results on the VSB100 and Moseg video segmentation benchmarks. We show in green and red present and missing trajectory entries, respectively. In the depth image red is close, blue is far. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

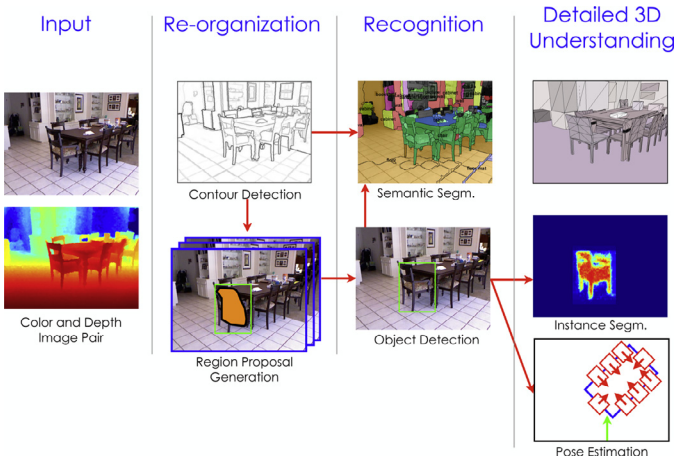

Fig. 18. Pipeline to represent objects in RGB-D images with corresponding models from a 3D CAD model library.

Fig. 19. Model placement: color image, placed models projected onto color image, depth map of rendered models (blue is closer, red is farther away). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

8. Putting it together

In our recent work [35], we have brought concepts presented in this paper together to automatically being able to represent objects in RGB-D images with corresponding models from a 3D CAD model library. We achieve this using the pipeline illustrated in Fig. 18, where we first detect and segment object instances (Section 6), and estimate their coarse pose [35] to initialize a search over a small set of 3D models, their scales and precise placements. Fig. 19 shows some sample results, where we can see that we can successfully address this problem in cluttered real world scenes.

Analogous work for RGB images is presented by [48]. We first detect and approximately segment objects in images using the approach from [37] (Section 3). Then, as outlined in Section 4, for each detected object we estimate coarse pose for using the approach from [77]. We use the predicted object mask and coarse pose to fit the top down deformable shape models that were learned from data. Finally, we recover high frequency shape details using low level shading cues by conditioning the intrinsic image algorithm of [5] on our inferred coarse 3D shape. We present the first fully automatic reconstructions on the PASCAL VOC dataset [18] and state-of-the-art object reconstruction as benchmarked on the PASCAL3D+ dataset [86]. Example results of reconstruction from a single image are shown in Fig. 20.

Note that these two systems are merely examples of how the 3R's of reconstruction, reorganization and reconstruction can jointly work towards the grand goal of scene understanding. Probabilistic graphical models or neural network architectures with multiple loss terms are both frameworks which can support such synthesis efforts. Over the next few years, we may expect systems to keep improving in accuracy at benchmarks such as MS COCO which emphasize detection and localization of individual objects, as well as other benchmarks which emphasize fidelity of geometric reconstruction.

But this is not all. Understanding a scene at the level of individual objects and their geometry is not the only goal of computer vision. Objects do not exist in isolation, they are in contextual relationships with each other and to humans and other agents that act in these environments. We cannot claim to understand a scene

Fig. 20. Fully automatic reconstructions obtained from just a single image using our system described above. Each panel (from left to right) shows the image with automatic detection and segmentation, the 3D model inferred using our method 'replaced in place' and the final depth map obtained.

fully until we have the ability to parse these relationships, make predictions about how the scene might evolve, and use our understanding of the scene to manipulate it or to navigate in it. Visual perception is not an end in itself, it connects to motor control as well as to general cognition. The progress that we have made so far provides an excellent substrate on which to build these connections.

Acknowledgments

This work was supported by ONR SMARTS MURI N00014-09-1-1051, ONR MURI N00014-10-10933 and NSF Award IIS-1212798.

References

[1] J. Aggarwal, M. Ryoo, Human activity analysis: a review, ACM Comput. Surv. (2011).
[2] I. Akhter, Y. Sheikh, S. Khan, T. Kanade, Trajectory space: a dual representation for nonrigid structure from motion, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011).
[3] P. Arbeláez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2011).
[4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, in: Proceedings of the CVPR, 2014.
[5] J.T. Barron, J. Malik, Shape, illumination, and reflectance from shading, IEEE Trans. Pattern Anal. Mach. Intell. (2015).
[6] E. Borenstein, S. Ullman, Class-specific, top-down segmentation, in: Proceedings of the ECCV, 2002.
[7] C. Bregler, A. Hertzmann, H. Biermann, Recovering non-rigid 3D shape from image streams, in: Proceedings of the CVPR, 2000.
[8] T. Brox, J. Malik, Object segmentation by long term analysis of point trajectories, in: Proceedings of the ECCV, 2010.
[9] S. Burer, R.D.C. Monteiro, Local minima and convergence in low-rank semidefinite programming, Math. Program. 103 (3) (2005) 427–444, doi:10.1007/s10107-004-0564-1.
[10] H.E. Burton, Theoptics of Euclid, J. Opt. Soc. Am. 35 (5) (1945) 357.
[11] R. Cabral, F. de la Torre, J. Costeira, A. Bernardino, Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition, in: Proceedings of the ICCV, 2013, doi:10.1109/ICCV.2013.309.
[12] J. Carreira, C. Sminchisescu, CPMC: automatic object segmentation using constrained parametric min-cuts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1312–1328.
[13] J. Carreira, S. Vicente, L. Agapito, J. Batista, Lifting object detection datasets into 3D, IEEE Trans. Pattern Anal. Mach. Intell. (2015).
[14] Y. Dai, H. Li, M. He, A simple prior-free method for non-rigid structure-from-motion factorization, in: Proceedings of the CVPR, 2012, pp. 2018–2025.
[15] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the CVPR, 2005.
[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of the CVPR, 2009.
[17] P. Dollár, C.L. Zitnick, Fast edge detection using structured forests, CoRR (2014). abs/1406.5549.
[18] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The Pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010).
[19] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013).
[20] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. (2010).
[21] S. Fidler, R. Mottaghi, A. Yuille, R. Urtasun, Bottom-up segmentation for top-down detection, in: Proceedings of the CVPR, 2013.
[22] K. Fragkiadaki, M. Salas, P. Arbelaez, J. Malik, Grouping-based low-rank trajectory completion and 3D reconstruction, in: Proceedings of the NIPS, 2014.
[23] F. Galasso, N.S. Nagaraja, T.J. Cardenas, T. Brox, B. Schiele, A unified video segmentation benchmark: annotation, metrics and analysis, in: Proceedings of the ICCV, 2013.
[24] R. Garg, A. Roussos, L. de Agapito, Dense variational reconstruction of non-rigid surfaces from monocular video, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23–28, 2013, pp. 1272–1279, doi:10.1109/CVPR.2013.168.
[25] A. Ghodrati, M. Pedersoli, T. Tuytelaars, Is 2D information enough for viewpoint estimation? in: Proceedings of the British Machine Vision Conference, vol. 2, BMVA Press, 2014, p. 6.
[26] J.J. Gibson, The Perception of the Visual World.1950.
[27] R. Girshick, Fast R-CNN, arXiv:1504.08083v1[cs.CV]2015.
[28] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the CVPR, 2014.
[29] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2015).
[30] R. Girshick, P. Felzenszwalb, D. McAllester, Discriminatively trained deformablepart models, release 5, http://www.cs.berkeley.edu/~rbg/latent-v5/.
[31] G. Gkioxari, J. Malik, Finding action tubes, in: Proceedings of the CVPR, 2015.
[32] C. Gu, J.J. Lim, P. Arbeláez, J. Malik, Recognition using regions, in: Proceedings of the CVPR, 2009.
[33] S. Gupta, P. Arbeláez, R. Girshick, J. Malik, Indoor scene understanding with RGB-D images: bottom-up segmentation, object detection and semantic segmentation, Int. J. Comput. Vis. (2014).
[34] S. Gupta, P. Arbeláez, J. Malik, Perceptual organization and recognition of indoor scenes from RGB-D images, in: Proceedings of the CVPR, 2013.
[35] S. Gupta, P.A. Arbeláez, R.B. Girshick, J. Malik, Aligning 3D models to RGB-D images of cluttered scenes, in: Proceedings of the CVPR, 2015.
[36] S. Gupta, R. Girshick, P. Arbeláez, J. Malik, Learning rich features from RGB-D images for object detection and segmentation, in: Proceedings of the ECCV, 2014.
[37] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014.
[38] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Hypercolumns for object segmentation and fine-grained localization, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2015.
[39] R. Hartley, R. Vidal, Perspective nonrigid shape and motion recovery, in: D. Forsyth, P. Torr, A. Zisserman (Eds.), Computer Vision ECCV 2008, Lecture Notes in Computer Science, vol. 5302, Springer Berlin Heidelberg, 2008, pp. 276–289, doi:10.1007/978-3-540-88682-222.
[40] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of the ECCV, 2014.
[41] H.V. Helmholtz, Physiological optics, Opt. Soc. Am. 3 (1925) 318.
[42] D. Hoiem, A. Efros, M. Hebert, Geometric context from a single image, in: Proceedings of the CVPR, 2005.
[43] J. Hosang, R. Benenson, P. Dollár, B. Schiele, What makes for effective detection proposals?, arXiv:1502.05082v1[cs.CV], 2015.
[44] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. 160 (1) (1962).
[45] A. Humayun, F. Li, J.M. Rehg, RIGOR: reusing inference in graph cuts for generating object regions, in: Proceedings of the CVPR, 2014.
[46] M. Jain, J. Gemert, H. Jegou, P. Bouthemy, C.G.M. Snoek, Action localization with tubelets from motion, in: Proceedings of the CVPR, 2014.
[47] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, M. Black, Towards understanding action recognition, in: Proceedings of the ICCV, 2013.
[48] A. Kar, S. Tulsiani, J. Carreira, J. Malik, Category-specific object reconstruction from a single image, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2015.
[49] P. Krähenbühl, V. Koltun, Geodesic object proposals, in: Proceedings of the ECCV, 2014.
[50] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks., in: Proceedings of the NIPS, 2012.
[51] E. Kruppa, Zur Ermittlung eines Objecktes aus zwei Perspektiven mit innerer Orientierung, Sitz.-Ber. Akad. Wiss., Wien, Math. Naturw. Kl. Abt. IIa 122 (1913) 1939–1948.
[52] T. Lan, Y. Wang, G. Mori, Discriminative figure-centric models for joint action localization and recognition, in: Proceedings of the ICCV, 2011.
[53] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. (1989).

[54] J. Long, E. Schelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the CVPR, 2015.
[55] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. (2004).
[56] M. Maire, S.X. Yu, P. Perona, Reconstructive sparse code transfer for contour detection and semantic labeling, in: Proceedings of the Asian Conference on Computer Vision (ACCV), 2014.
[57] J. Malik, Visual grouping and object recognition, in: Proceedings of the 11th International Conference on Image Analysis and Processing, 2001, IEEE, 2001, pp. 612–621.
[58] D. Marr, Vision: A Computational Approach, 1982.
[59] C. Nandakumar, A. Torralba, J. Malik, How littledo we need for 3-D shape perception? Perception 40 (3) (2011) 257.
[60] B. Pepik, M. Stark, P. Gehler, B. Schiele, Teaching 3d geometry to deformable part models, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3362–3369.
[61] M. Pollefeys, R. Koch, M. Vergauwen, L.V. Gool, Metric 3D surface reconstruction from uncalibrated image sequences, in: 3D Structure From Multiple Images of Large Scale Environments, in: LNCS, Springer-Verlag, 1998, pp. 138–153.
[62] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. (2010).
[63] X. Ren, J. Malik, Learning a classification model for segmentation, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, IEEE, 2003, pp. 10–17.
[64] L.G. Roberts, Machine Perception of Three-dimensional Solids, Technical Report 315, MIT Lincoln Laboratory, 1963.
[65] M.D. Rodriguez, J. Ahmed, M. Shah, Action mach: a spatio-temporal maximum average correlation height filter for action recognition, in: Proceedings of the CVPR, 2008.
[66] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, A. Zisserman, Using multiple segmentations to discover objects and their extent in image collections, in: Proceedings of the CVPR, 2006.
[67] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, OverFeat: integrated recognition, localization and detection using convolutional networks, in: Proceedings of the ICLR, 2014.
[68] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of the CVPR, 2013.
[69] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, in: Proceedings of the ECCV, 2012.
[70] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Proceedings of the NIPS, 2014.
[71] K. Simonyan, A. Zisserman, Very deepconvolutional networks for large-scale image recognition, arXiv:1409.1556. 2014.
[72] Y. Tian, R. Sukthankar, M. Shah, Spatiotemporal deformable part models for action detection, in: Proceedings of the CVPR, 2013.
[73] J. Tighe, M. Niethammer, S. Lazebnik, Scene parsing with object instances and occlusion handling, in: Proceedings of the CVPR, 2014.
[74] K.-C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pac. J. Optim. (2010).
[75] C. Tomasi, T. Kanade, Shape and Motion From Image Streams: A Factorization method, Technical Report, IJCV, 1991.
[76] L. Torresani, A. Hertzmann, C. Bregler., Non-rigid structure-from-motion: estimating shape and motion with hierarchical priors, IEEE Trans. Pattern Anal. Mach. Intell. (2008).
[77] S. Tulsiani, J. Malik, Viewpoints and keypoints, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2015.
[78] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. (2013).
[79] R. Vaillant, C. Monrocq, Y. LeCun, Original approach for the localisation of objects in images, IEE Proc. Vis. Image Signal Process. (1994).
[80] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the CVPR, 2001.
[81] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the ICCV, 2013.
[82] L. Wang, Y. Qiao, X. Tang, Video action detection with relational dynamic-poselets, in: Proceedings of the ECCV, 2014.
[83] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for generic object detection, in: Proceedings of the ICCV, 2013.
[84] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Comput. Vis. Image Underst. (2011).
[85] M. Wertheimer, Laws of organization in perceptual forms, A SourceBook of Gestalt Psychology (1923).
[86] Y. Xiang, R. Mottaghi, S. Savarese, Beyond Pascal: a benchmark for 3D object detection in the wild, in: Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2014, pp. 75–82.
[87] Y. Yang, S. Hallman, D. Ramanan, C.C. Fowlkes, Layered object models for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012).
[88] C.L. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges, in: Proceedings of the ECCV, 2014.