The Image Torque Operator: A New Tool for Mid-level Vision

Morimichi Nishigaki
University of Maryland
College Park, MD 20742
michi@cs.umd.edu

Cornelia Fermüller
University of Mayland
College Park, MD 20742
fer@cfar.umd.edu

Daniel Dementhon
Applied Physics Lab,
Laurel, Maryland
Daniel.DeMenthon@jhuapl.edu

Abstract

Contours are a powerful cue for semantic image understanding. Objects and parts of objects in the image are delineated from their surrounding by closed contours which make up their boundary. In this paper we introduce a new bottom-up visual operator to capture the concept of closed contours, which we call the 'Torque' operator. Its computation is inspired by the mechanical definition of torque or moment of force, and applied to image edges. The torque operator takes as input edges and computes over regions of different size a measure of how well the edges are aligned to form a closed, convex contour. We explore fundamental properties of this measure and demonstrate that it can be made a useful tool for visual attention, segmentation, and boundary edge detection by verifying its benefits on these applications.

1. Introduction

A problem central to visual scene interpretation is the localization and segmentation of objects and detection of their boundaries in the visual scene. All visual processing starts with the cluttered two-dimensional image(s), which are formed as the projections of the three-dimensional world. Psychologists of the early twentieth century. argued that human vision organizes the image clutter at the early stages of interpretation through a process of figure ground segmentation by identifying the image regions which are object-related for further processing. They suggest that certain principles are applied to group pieces of image and locate borders of figures. Most theorists of Vision will argue that scene interpretation involves processes at different levels of abstraction, which are categorized into low, mid, and high level vision. Low level processes compute features, such as local edges, color, texture and image motion. Mid-level processes combine the features into larger coherent patches to obtain surfaces, and other information such as 3D motion and lighting, and high level processes utilize semantic information to recognize objects, actions and scenes. Many will agree that mid-level vision to a large degree is about implementing organizational principles, some of which have been proposed by the Gestalt theorists.

Few studies have explored mid-level cues as a tool for boundary detection and segmentation. Most studies start with clean, segmented contours to explore cues such as junctions [24], parallelism, line continuity [32] and convexity [4, 39, 11]. To deal with cluttered images, recently the focus has shifted to data-driven approaches [34]; for example [38] learns mid-level cues from low level cues. Segmentation approaches are usually based on local cues (edges and intensity) as input to a global optimization, but recently many methods first compute super-pixels [35] by over-segmenting the image into perceptual uniform regions based on the statistics in neighborhoods or affinity between points.

In this paper we introduce a new semi-global operator of mid-level vision called the 'torque', that captures the organizational concept of 'closure'. This operator is defined on oriented edges and provides a measurement of how well the local edges in a patch fit a contour. Its function is to provide a tool for collecting edge information from regions of different extent in order to enforce edges that form boundaries and discard edges due to texture. The reader can skip ahead to Fig. 7 to get an intuition for a map computed with the torque operator.

After a description of related work, we present an analysis and description of the properties of this operator. Then we apply the operator to a few problems of image processing on single images, specifically the problems of boundary detection, visual attention, and foreground segmentation, in order to verify its usefulness as a tool that can improve existing techniques. Finally, we conclude with a discussion on extending this operator to multiple view data, where contour orientation can be defined on the basis of geometric information (motion and depth).

2. Related Work

Segmentation involves combining information from the interior of surfaces which are smooth in some quantities, such as color, intensity, texture [1], motion [8, 30, 12], depth [22], and the discontinuities in these quantities, which

are reflected as luminance edges and discontinuities in the texture and the geometric cues. Most approaches treat the problem of segmentation as dividing the image into multiple regions using different clustering techniques [10, 7]. Our focus here is to segment objects, and methods closer related to this concept are those that consider the problem of segmentation as separating one foreground object from background. The problem is usually modeled as optimizing for a binary labeling that assigns each pixel a label such that the labeling is both consistent with the observed data and piecewise smooth. One sets up an energy model, which either is formulated probabilistically and solved using belief propagation or graph cut methods [22], or continuous and reformulated with differential equations and solved using active contours and variational approaches [6, 29]. An interesting biological motivated formulation was introduced in [28], which segments in the polar coordinate system by minimizing for a closed contour surrounding a fixation point using a graph cut formulation defined on edges only. However, all approaches have the problem of being biased, usually towards small regions with small and smooth boundary. This is because of texture edges, which in real images are always present, in conjunction with minimizations biased towards certain shapes. For example graph cuts [22] are known to favor small areas, the polar coordinate representation favors circular blobs, and variational minimizations [37] explicitly minimize the length and/or smoothness of the bounding contour. The torque operator can help alleviate the bias, as it can be used to locate the regions surrounded by contours, locate contours of larger extent, and separate texture edges from boundary edges.

Contours are an important cue for segmentation and recognition. By the term contour, we generally refer to extended curve or edge fragments which present some meaningful geometric concept. As a tool for recognition, so called contour patches, which are local descriptors of spatial edge distributions have been developed. For example, the shape context descriptor [5] encodes the spatial distribution of edge points in log polar space, or [21] defines a feature detector based on the saliency of local convexity. Recently, data driven approaches have become popular, which acquire contour fragments and their detectors [23, 31] using sophisticated learning techniques from large amounts of data.

In the context of segmentation, one is interested in the contours that form the boundaries which separate foreground from background. Great advances in contour detection have been achieved through data-driven approaches, championed in the work of Martin *et al*. [27]. In this work, local cues, such as brightness, color, texture, and their gradients are combined, and weights for each cue are learned using image data sets with ground truth contours [27, 25]. In similar spirit [13] learns edge classifiers from simple features in image patches, Ren [33] combines information of local operators from multiple scales, and the high performance contour detection algorithm in [3] includes a globalization process to combine local edges based on the affinity of distant pixels. While all these techniques are based on learning approaches, the torque mechanism achieves contour localization in a bottom-up fashion. The operator could, however, also be used as input to a learning approach.

Visual attention or saliency can be used as pre-process of foreground segmentation. Attention mechanisms are classified into bottom-up and top-down processes. Top-down attention is more complex because it represents objects in memory [17] and uses the memory to detect likely objects in an attended visual scene. Bottom up attention is driven by low level processes. Probably the best known model of visual attention is the one proposed by Itti *et al*. [19]. In this model, first local feature maps are computed from color, intensity and orientation information as the difference of Gaussian filters at multiple scales, which approximate the 'center-surround' differences of neurons. Larger center surround differences are considered more 'conspicuous'. Then a combined saliency map is constructed by adding the normalized feature maps. Related approaches differ in the choice of feature vectors and combination of features. In our experimental section we will compare against the method of Harel *et al*. [16], which computes a saliency map based on the dissimilarity of features in regions using a graph-based approach. Harel *et al*. evaluated the performance of their detector on its ability to predict human attention using the human fixation data of Einhäuser *et al*. [14], and reported to achieve 98% of the ROC area of human based control, while the model by Itti *et al*. achieved only 84%. Recent works in fixation and attention [18, 15] offers an alternative to the traditional "early" feature saliency theories. Based on systematic psychophysical experiments [18] suggests that observers look at objects as if they knew them before they became aware of their identity, and [15] shows that the hypothesis that humans attend to objects has a better predictive power in the data than any other theory. In our paper we will use the torque measure to derive a saliency map for visual attention. The extrema of the torque measure often appear at the locations in the image where objects are surrounded by edges. Thus the torque appears to be a good measure to model object driven visual attention.

3. Torque Measure

3.1. Definition

Torque is a measure used in physics to express how much a force acting on an object causes that object to rotate. It is defined as:

$$\vec{\tau} = \vec{r} \times \vec{F}, \tag{1}$$

where τ is the torque vector, $\vec{r}$ is the displacement vector from the axis of rotation to the point where the force is applied, and $\vec{F}$ is the force vector. Based on this concept we define a torque measure for images. It is treated as if a force is applied at a point on an image edge along the tangent of the edge. For an arbitrary point, which we call the center point, the rotation axis for the torque is imagined as if it is going through that point in three-dimensional space. Then, we can measure the torque at any point in the image with respect to the rotation axis as illustrated in Figure 1. Since the displacement vector and the force vector are on the image surface, the torque vector is always perpendicular to the image surface. Therefore, we call the value of the torque vector along the axis perpendicular to the image simply the *torque* or *torque value* hereafter.
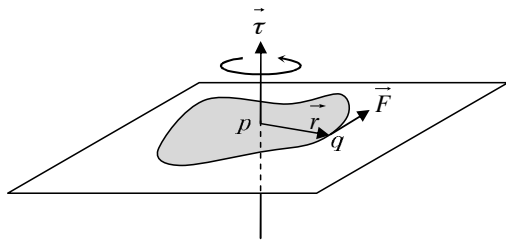

Figure 1. Torque for images: Consider a center point p and an edge point q, to which we assign a force vector $\vec{F}$ along the tangent of the edge. Denoting as $\vec{r}$ the vector from p to q, the torque vector at q is defined as $\vec{\tau} = \vec{r} \times \vec{F}$. Its value along the axis perpendicular the image will be called the *torque*.

In general, an image edge is represented by a function $C \in \mathbb{R}^2$ with parameter t as follows:

$$C(t) = (C_x(t), C_y(t))^T. \tag{2}$$

The derivative of C is treated as the force applied at an edge point. Thus we can define *the torque for points on edge curves* as:

$$\tau_p(t) = (C(t) - p) \times C'(t), \tag{3}$$

where p is the point where the rotation axis pierces the image. For simplicity, the cross product of two dimensional vectors is obtained by cross-multiplying the vectors. (This is equivalent to adding a zero as a third element to the two dimensional vectors and considering only the third component of the cross product of such obtained 3D vectors.) Figure 2 gives an illustration of the geometric relation of the two vectors in eq. (3). Based on the definition of the torque for a point, we define the *torque of an image patch* as the sum of the torque values of all edge points in the image patch. A rigorous definition of the torque of an image patch of arbitrary shape is as follows:

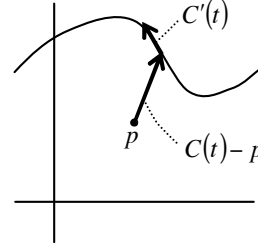$$\tau_P = \sum_i^N \sum_j^{M^i} \frac{1}{2|P|} \int_{t_0^{i,j}}^{t_1^{i,j}} \tau_p^i(t) \, dt, \tag{4}$$

Figure 2. Torque for point on edge curve

$$(\ )$$

where P denotes an image patch, and p is a point inside the patch. $|P|$ is the area of the patch, N is number of edge curves overlapping the image patch, and $\tau_p^i(t)$ is the torque at value t of the i-th edge curve. The i-th edge curve has M^i segments within the image patch, and the j-th segment of the i-th edge curve is between value $t_0^{i,j}$ and $t_1^{i,j}$ in parameter t.

Since our images are discrete, edges are represented by a set of pixels in the image. Then, the force is defined by an oriented unit vector in the direction of the edge, and the *torque value of an edge pixel* amounts to:

$$\tau_{pq} = \|\vec{r}_{pq}\| \sin \theta_{pq}, \tag{5}$$

where p is the point on the image through which the rotation axis passes. q is an edge point. $\vec{r}_{pq}$ is the displacement vector from p to q, and θ is the angle between the displacement vector and the tangent vector.

Note that in our definition edges are oriented. Thus, the value of the torque can have positive and negative values. If multiple images are available we can define the orientation on the basis of depth. If we have a single image only, we define it on the basis of intensity, as perpendicular clockwise to the image gradient, such that the brighter side is on the right of the edge and the darker side on the left.

Similarly to the definition on edge curves, the discrete *torque of an image patch* is defined as:

$$\tau_P = \frac{1}{|P|} \sum_{q \in E(P)} \tau_{pq}, \tag{6}$$

where $E(P)$ is a set of edge points in the patch P, and p is a point inside the patch.

In principle the shape of the patch could be arbitrary, but in this paper we will use either disk or square patches, and p will be the center of the disk or square.

3.2. Properties of the torque

Next we discuss how the value of the torque is related to the contours in the image patch. We describe some properties of the torque in order to motivate how the torque will be used.

Torque and Area: Since the torque is defined by the cross product of vectors, it is essentially related to the area defined by these vectors as shown in figure 3. This relationship can be easily extended to edge curves. Assuming the edges are clean continuous curves, the magnitude of the torque in a patch is related to the position of the curves in the patch and their shape. The torque of a closed curve, completely inside the patch, is proportional to its area. Taking the normalization factor in definition of torque into account, the closer the patch boundaries surround the curve, the larger the torque. For curve segments intersecting the boundary of the patch with center p at two intersection points q_1 and q_2, the torque is proportional to the area enclosed by the edge curve and the two line segments pq_1 and pq_2. This is depicted in figure 4.
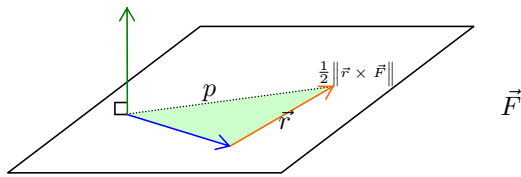

Figure 3. Cross product and area: The triangle enclosed by vector $\vec{r}$ and $\vec{F}$ is equivalent to $\left\| \vec{r} \times \vec{F} \right\| / 2$.
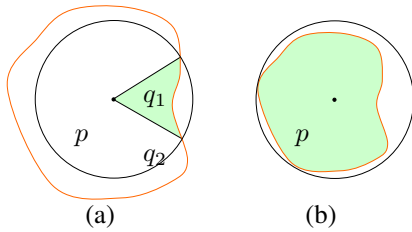

Figure 4. Relationship between torque and area: Two cases are shown: (a) the disk patch is smaller than the object and covers only part of the object boundary. (b) the disk patch covers the whole object boundary.

Texture vs Boundary: The torque will be larger when it contains extended contours and will be largest if the contours are closed. On the other hand, the torque will be small if the edges are random. Thus, intuitively texture edges lead to small torque values and boundary edges to large values.

Extrema in Torque: The torque value tends to be large in magnitude if the patch contains extended contours close to the boundaries of the patch. Therefore, it is expected that the torque measure can be used to find the locations in the image where edges are structured, that is, forming extended contours. Furthermore, it will give us the scale of the region of these edges. Figure 5 gives an example to illustrate this property. Referring to the figure, the torque was computed at every point in the image using disk patches of four dif-

ferent sizes. The location of the structured edges, *i.e.* the triangle, can be inferred from the extremum of the torque value over space and patch sizes. The patch size of the extremum of the torque indicates the size of the triangle. Figure 6 gives another example, this time for a circular contour. The torque values were computed at the center of the circle using disk patches with diameter in the range of 1 to 120. The extremum in the plot of the torque values over scale indicates the size of the circle. From these examples it can be seen that the extremum of torque indicates the existence of structured edges, and the corresponding patch size indicates the scale of the structured edges. Furthermore, it is worth noting that the torque value tends to be of same sign inside closed contours, and of opposite sign outside. The sign depends on whether the edge direction is clock-wise or counter-close-wise.
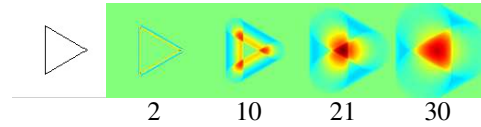

Figure 5. Torque maps for a triangle computed over four patch sizes: Torque values are color coded with red color assigned to positive and blue to negative torque values. Dark red corresponds to high positive torque values. The numbers below the torque maps denote the radius of the disk patch in pixels. The image is of size 101×101 pixels, and the height of the triangle is 40 pixels.
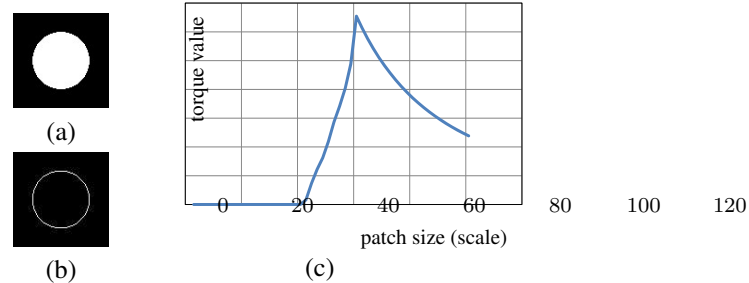

Figure 6. Torque values for a circle computed over a range of patch sizes: (a) is the example image, and (b) is the edge map of the image. The circle is located in the middle of image, and its diameter is 61 pixels. The torque values are computed at the center point of the image over different patch sizes. Plot (c) shows that the maximum torque at the center of image is obtained when the patch tightly circumscribes the circle.

3.3. Representation of the torque

First, let us get an intuition for the torque on a real image. Referring to the figuree 7, for the image on the upper left from the Berkeley data set, the torque map was computed using patches of three sizes (5×5, 21×21 and 45×45), *i.e.* for every pixel p we computed the torque with patches of different sizes and centered at p. We used the color coding

explained in the second row. Pixels of blue color correspond to negative torque value, and pixels of red color correspond to positive torque value. The negative values originate from dark regions surrounded by lighter background and the positive values from bright regions surrounded by dark values. As one can see from the figure, for small patch sizes, the large torque values are near the image edges, and as the size of the patch increases the larger torque values move to the center of object regions. We then define data structures that combines all patch sizes, which can be described by the following equations:

$$V(x,y) = \tau(x, y, \hat{s}(x,y)), \quad (7)$$

$$S(x,y) = \operatorname{sgn}(V(x,y)) \cdot \hat{s}(x,y), \quad (8)$$

$$\hat{s}(x,y) = \operatorname*{argmax}_{s} |\tau(x,y,s)|. \quad (9)$$

where $\tau(x,y,s)$ is the torque value at point (x,y) with patch size (scale) s. We call this three-dimensional volume of τ the *torque volume*, and V and S the *torque value map* and *scale map*, respectively. The torque value map, at every pixel codes the value of largest absolute value over all scales, and the scale map codes the corresponding scale. Referring to figure 7, the second row shows the torque value map of the image. Looking at this structure, it becomes intuitively clear that the torque can be used as a mid level tool. Large (positive and negative) values of torque at a point indicate the likelihood of large structures around it. This makes this structures useful as basis for an attention mechanism. The edges contributing to large values in the torque map are likely parts of extended contour, and we thus can use the torque mechanism for boundary localization. Then these boundaries in addition with the torque value can be used for segmentation.
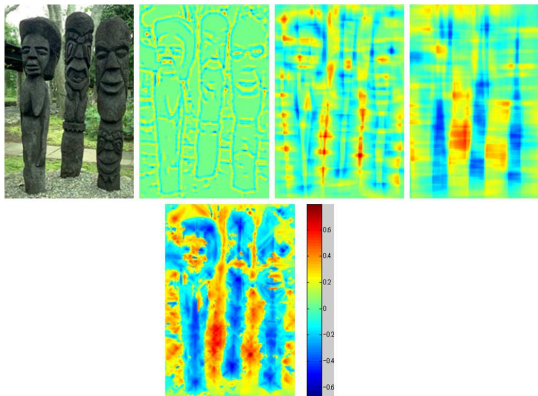

Figure 7. Illustration of the torque: Upper row: For the image on the upper left, the torque is computed with three patches of increasing size. Lower row: Combination of the torque values from all patch sizes into one map, which we call the value map.

4. Application

It is reasonable to expect, on the basis of current knowledge, that the visual processing necessary to find an object in a scene consists of three modules: visual attention, boundary detection, and foreground segmentation, as depicted in Fig. 8. Such an active approach is especially well suited for mobile robot applications. The attention mechanism is important for focusing the processing to the conspicuous region - the region of interest. In biological systems the attention is indicated by the fixation point. Segmenting foreground from background then requires that we detect edges surrounding the fixation point. The torque mechanism can be used to link the edges into contours and bias the segmentation towards these contours. In this section, we evaluate how much the proposed torque operator can improve each of the three steps mentioned above by comparing it against other methods in standard database settings.
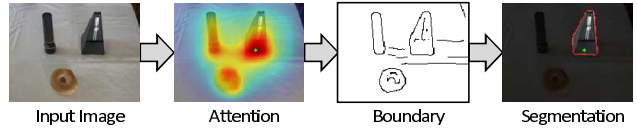

Figure 8. Visual processing using the image torque operator

4.1. Visual Attention

The torque measure tends to have extrema at points surrounded by boundary edges and at patch sizes corresponding to object size. This property is expected to be useful as a cue for bottom-up visual attention. In our experiment we computed two torque-based saliency maps. The first is generated as a mixture of Gaussians, with the Gaussian kernels centered at the extremum values in the torque volume. The second is the weighted sum of the first saliency map and the graph-based visual saliency (GBVS) by Heral *et al*. [16]. As weights we used 0.3 for the torque-based saliency map and 0.7 for the GBVS, where the weight values were found empirically from a few trials using a small subset of the dataset. The computed torque-based saliency maps were normalized in the range $[0, 1]$.

The torque-based saliency maps were compared quantitatively against the saliency maps of Itti *et al*. [19] and Harel *et al*. using the eye tracking data by Judd *et al*. [20] to generate ground truth saliency maps. For all methods, fixation points were extracted from the data, and saliency maps were generated as mixture of Gaussian distributions with the Gaussians centered at the fixation point, and the maps were normalized.

The ground truth saliency maps were binarized by a threshold (of value 0.5). To compute the ROC curves for the different methods, we used a set of thresholds equally distant in $[0, 1]$. For each threshold the saliency map was binarized. We then measured precision and recall of the

such estimated saliency maps as follows:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{FP + FN}, \quad (10)$$

$$TP = |S \cap \mathcal{G}|, FP = |S \cap \overline{\mathcal{G}}|, FN = |\overline{S} \cap \mathcal{G}|, \quad (11)$$

where S is the computed saliency map binarized by a threshold, and $\mathcal{G}$ is the binarized ground truth saliency map. P and R denote precision and recall, and TP, FP, FN denote true positive, false positive, and false negative, respectively.

We resized the test images so that the length of the shorter side of the images was 150 pixels. The standard deviations of the Gaussian distributions used to generate the ground truth and torque-based saliency maps, were both set to 25 pixels.

Fig. 9 shows the ROC curves and maximum F-measures computed from the 898 test images in the dataset. Examples of the computed saliency maps along with the ground-truth are shown in Fig. 10.

The quantitative comparison shows that the torque-based attention by itself doesn't outperform GBVS. This does not come as a surprise, as GBVS is a sophisticated method that includes a series of features, including texture, and it codes relations between different parts of the image. Attention is related to recognition, and texture plays a very important role in recognition and has been shown by current Computer Vision applications to be more powerful than contour. However, the torque measure as an additional mid-level visual cue improves the quality of GBVS.
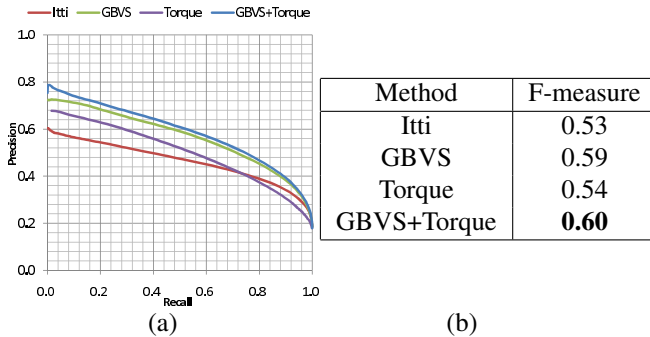

Method	F-measure
Itti	0.53
GBVS	0.59
Torque	0.54
GBVS+Torque	**0.60**

(a)   (b)

Figure 9. Evaluation of attention models. (a) ROC curves. (b) F-measure.

4.2. Boundary Detection

The contribution of edge points to torque values of different patches can be written as follows:

$$v_q = \sum_{\{P|q \in P\}} \tau_{p(P)q}, \quad (12)$$

where $p(P)$ denotes the center of the patch P, and q denotes an edge point. Extrema of the torque value in the
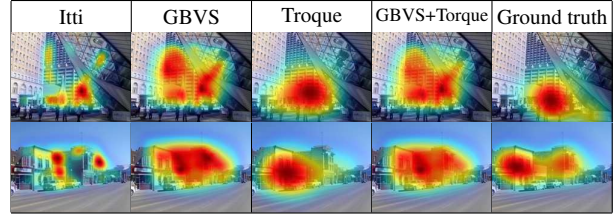

Figure 10. Visual attention results for two test images. The saliency maps obtained from four different methods and the ground-truth are visualized by overlaying them on the test images.

torque volume indicate the existence of edges surrounding the center of the patch at the corresponding scale. Therefore, an edge point's contribution in eq. (12) to an extremum is expected to be high for structured edges corresponding to extended contours. In our method, edges are strengthened by combining the original edges with this contribution value as follows:

$$d_s = \frac{1}{1 + e^{-(c_0 + c_1 d_o + c_2 d_t)}}, \quad (13)$$

where d_o is the original edge intensity, and d_t is the normalized torque contribution. c_0, c_1, and c_2 are constant parameters, that were learned using training images from the Berkeley dataset. The such strengthened edges are expected to represent object boundaries. Examples are shown in Fig. 11. Edges were computed using the Canny edge detector as shown in (b), which were then used to compute the torque. As can be seen from (c), the strengthened edges tend to be stronger at boundary edges of objects, and weaker at texture edges.

We used the Berkeley dataset [26] to compare quantitatively the improvement of boundary detection by the torque. While the Canny edge method scores 0.57, the torque-based edge strengthening method increases the score to 0.59 in the F-measure of the Berkeley benchmark. We also tested with the state-of-art boundary detection method known as gPb [3]. The improvement by combining our method with the proposed torque operator was about 0.2% in F-measure. The improvement was very minor. However, this is expected, as the torque is not meant to localize general salient edges, but to detect closed contours. On the other hand, the gPb method, which integrates higher level semantic information about contours from learning with visual cues in a globalization step, is designed for this task.

4.3. Segmentation

The strengthened edges are expected to be useful for figure ground segmentation, because object boundaries are emphasized. It is not difficult to imagine various ways of using the torque information in different techniques to bias the segmentation towards the edges. Here we simply uses the strengthened edges (Sec. 4.2) in an edge-based graph-cut algorithm.
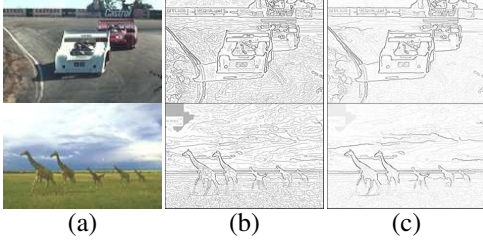
Figure 11. Examples of Strengthened Edges: (a) Test images. (b) Canny edges. (c) Strengthened edges. The Canny edges shown in (b) are used to compute strengthened edges in (c).

For a quantitative evaluation of figure-ground segmentation, we used the dataset by Stein *et al.* [36]. For each reference image in the data set we selected a single foreground object, and used the centroid of the object as a fixation point. Then we applied the fixation based segmentation algorithm by [28]. This algorithm separates foreground from background using a graphcut on a probability map of edges in a polar coordinate system . We compared four edge maps using different edge detection methods in the graph-cut segmentation: the Canny edge map, the boundary probability map (Pb) by Martin *et al.* [27], a strengthened edge map based on Canny edges, and a strengthened edge map based on Pb edges. In addition, we also compared with the non-edge based level-set segmentation by Chan and Vese [9]. A small rectangular region centered at the given fixation point was used as seed region in this method. To strengthen the edges we used directly the normalized torque contribution d_t in eq. (13). The quality of the segmentation was evaluated by the segmentation covering defined as follows [2]:

$$C = \frac{|S \cap \mathcal{G}|}{|S \cup \mathcal{G}|},$$ (14)

where S and $\mathcal{G}$ are the computed segmentation and the ground truth segmentation, respectively. Both S and $\mathcal{G}$ are represented by binary labels.

Table 1 shows the results of quantitative comparison. For each visual cue the average covering over 28 test images is given. The label 'Torque' in the table refers to the edge strengthening with the torque using Canny edges in the left part of the table, and Pb edges in the right part of the table. As can be seen, introducing the torque measure as a mid-level visual cue significantly improves the segmentation. We can also see from the performance of the Chan-Vese method, that the segmentation of objects for this data set, given only the fixation point, is a challenging task. Examples of segmentation results are shown in Fig. 12.

5. Conclusion

We introduced the "Torque Operator", a new mid-level visual operator, and explored some of its fundamental properties, using both theory and experiments. The basic tenet of the torque operator is that it tends to generate large val-

Table 1. Covering of Foreground Segmentation

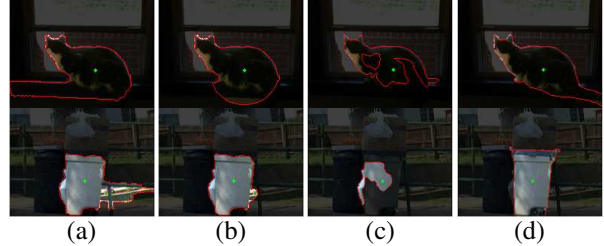| Visual Cue	Covering	Visual Cue	Covering
Canny	0.32	Pb	0.40
Torque	**0.47**	Torque	**0.48**
		Chan-Vese	0.21

Figure 12. Examples of segmentations: (a) Segmentation using Canny edges. (b) Segmentation using strengthened edges by the torque. (c) Segmentation by Chan-Vese method. (d) Ground truth. The green dots denote the fixation points.

ues when edges are aligned in a way surrounding the center of the patch, and their scale matches the size of the patch. Therefore, torque extrema are expected to indicate contours surrounding a point, and this point often is the center of an object. These properties were utilized in three applications: attention, boundary detection, and segmentation. It was shown quantitatively that for all three applications the torque operator enhances the performance.

6. Extension and Outlook

We have only begun to scratch the surface of a new visual operator. Being between local operators (like the derivative operators) and global operators (like the multidimensional histogram of features), it has the potential to support many computer vision applications. In the experiments of this paper, we haven't utilized the benefit of torque operator in the most efficient way yet. Only strengthened edges were used in the experiments on segmentation, but the torque value itself also is expected to be a useful visual cue for segmentation. Furthermore, since the torque operator indicates the scale of the surrounding edge structure, *e.g.* objecthood, it also could be used in conjunction with segmentation to determine the region of attention. In this paper we applied the torque operator only to rectangular and circular patches. Further extension of this work could be using patches of other shapes such as elliptic patches, so that the torque operator could be adapted to elongated objects, for example. Another direction of extension is applying the torque operator to different cues. While we applied it to edges in single images in this paper, the torque operator can also be applied to geometric edges such as depth edges and motion edges, where we expect it to perform better for detecting objects.

References

[1] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*, 2007.

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

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

[4] E. Arkin, L. Chew, D. Huttenlocher, K. Kedem, and J. Mitchell. An efficiently computable metric for comparing polygonal shapes. *PAMI*, 13(3):209–216, 1991.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, April 2002.

[6] A. Blake and M. Isard. *Active Contours*. Springer, 2000.

[7] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[8] T. Brodsky, C. Fermüller, and Y. Aloimonos. Simultaneous estimation of viewing geometry and structure. In *ECCV*, pages 342–358, 1998.

[9] T. F. Chan and L. A. Vese. Active contours without edges. *PAMI*, 10(2):266–277, 2001.

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

[11] P. Corcoran, P. Mooney, and A. Winstanley. A convexity measure for open and closed contours. In *BMVC*, 2011.

[12] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR*, 2006.

[13] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *CVPR*, 2006.

[14] W. Einhäuser, W. Kruse, K.-P. Hoffmann, and P. König. Differences of monkey and human overt attention under natural conditions. *Vision Research*, 46(8-9):1194–1209, 2006.

[15] W. Einhäuser, M. Spain, and P. Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14):1–26, 2008.

[16] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006.

[17] A. Hollingworth, C. Williams, and J. Henderson. To see and remember: Visually specific information is retained in memory from previously attended objects in naturalscenes. *Psychonomic Bulletin and Review*, 8:761–768, 2001.

[18] L. Holm, J. Eriksson, and L. Andersson. Looking as if you know: Systematic object inspection precedes object recognition. *Journal of Vision*, 8(4):1–7, 2008.

[19] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, 1998.

[20] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, 2009.

[21] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, 2004.

[22] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bilayer segmentation of binocular stereo video. In *CVPR*, 2005.

[23] M. Kumar, P. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *BMVC*, 2004.

[24] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer, Boston, 1994.

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

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

[27] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[28] A. Mishra, C. Fermˇller, and Aloimonos. Active segmentation for robotics. In *IROS*, 2009.

[29] D. Mumford and J. Shah. Functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[30] A. Ogale, C. Fermüller, and Y. Aloimonos. Motion segmentation using occlusions. *IEEE T. PAMI*, 27(6):988–992, 2005.

[31] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006.

[32] P. Parent and S. Zucker. Trace inference, curvature consistency, and curve detection. *PAMI*, 11(8):823–839, 1989.

[33] X. Ren. Multi-scale improves boundary detection in natural images. In *ECCV*, 2008.

[34] X. Ren, C. C. Fowlkes, and J. Malik. Cue integration in figure/ground labeling. In *Advances in Neural Information Processing Systems*, 2005.

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

[36] A. N. Stein, T. S. Stepleton, and M. Hebert. Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In *CVPR*, 2008.

[37] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *IJCV*, 50(3):271–293, 2002.

[38] S. Zheng, A. Yuille, and Z. Tu. Detecting object boundaries using low-, mid-, and high-level information. *CVIU*, 114(10):1055–1067, 2010.

[39] J. Zunic and P. Rosin. A new convexity measure for polygons. *PAMI*, 26(7):923–934, 2004.