Exploiting Protrusion Cues for Fast and Effective Shape Modeling via Ellipses

Alex Wong[1]
alexw@cs.ucla.edu
Brian Taylor[1]
btay@cs.ucla.edu
Alan Yuille[2]
alan.yuille@jhu.edu

[1] Department of Computer Science
University of California, Los Angeles
[2] Department of Computer Science
Johns Hopkins University

Abstract

Modeling objects with a set of geometric primitives is a key problem in computer vision and pattern recognition with numerous applications including object detection and retrieval, tracking, motion and action analysis. In this paper, we attempt to represent 2D object shapes with a number of ellipses with semantic meaning (e.g. one ellipse may correspond to an arm, while two other ellipses may represent a bent leg), while maintaining a high coverage of the shapes. We propose a novel ellipse fitting method based on psychology and cognitive science studies on shape decomposition and show that our shape coverage compares well with the state of the art methods, while significantly outperforming them in run-time by as much as 508 times in our evaluation of the methods on over 4000 2D shapes. We also demonstrate the value of our method to higher-level processing via an example application in creating 3D ellipsoid models of PASCAL horses.

1 Introduction

Fitting ellipses to 2D shapes has many applications in object detection and retrieval [27, 28], tracking [12], motion and action analysis [1, 20]. For modeling 2D shapes, ellipses provide a compact representation of complex objects and their structure, as most objects can be broken down into a set of rigid parts. Specifically, ellipse fitting can serve as an intermediary step for modeling human body parts [9, 11], where each ellipse models a semantic component (e.g. arms, torso, head, legs). Given the ability of the state of the art semantic segmentation approaches [6, 16, 24, 25, 30] and motion boundary detection methods [2, 18, 21, 26] to produce reliable object masks, representing these masks as ellipse models becomes a natural next step for understanding the segmented objects' structures. We demonstrate an example application of this by combining the results of multiple 2D ellipse models to hypothesize a pseudo-3D ellipsoid model. Leveraging such models can facilitate detection and segmentation in images and video where objects are partially occluded or difficult to recognize.

Fitting a single ellipse to a set of pixels has well-studied solutions (e.g. Least Square Fitting [8] and the Hough Transform [32]). Fitting a set of ellipses to cover a pixel-wise mask, however, proves more difficult as one must consider the alignment of ellipses to the mask as well as the relationships between the ellipses. The task of 2D shape modeling using ellipses

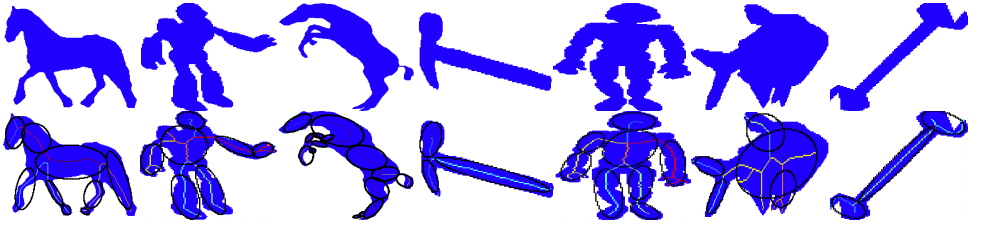

Figure 1: Examples of the output on four different datasets: LEMS, MPEG-7, SiSHA, PASCAL Horses. The top row denotes the given 2D shape and the bottom denotes our results. Thin lines of varying color indicate the symmetry axes of distinct parts. Black ellipses denote our output set of ellipses fit to the parts. Note: our method finds the locations of joints and models them with multiple ellipses (e.g. the limbs of the humanoids).

(Fig. 1) can be viewed as a difficult search problem as one must determine the number of ellipses n and estimate the parameters of the n-ellipses that approximate the 2D shape objects. We use the shape coverage of the ellipses[1] to define how well the model fits the shape. Trivially, fitting a large number of ellipses will provide good coverage; however, it will also increase the model complexity. We propose a method that achieves a balance between the model complexity and shape coverage. Inspired by psychology and cognitive science studies [2, 5, 10], our method produces parts that align well with the human perception of objects as compositions of parts. Since our work effectively yields a mid-level representation of 2D object parts, we demonstrate an application of our approach by constructing 3D ellipsoid models of PASCAL horses (Fig. 6) in Sec. 5.

We describe an overview of our method in Fig. 2. Like [19], we employ the 2D skeleton [3] as an initialization for our method. However, our method also considers the protrusion cues [33] of the 2D shape, with which we approximate the n-ellipse regions. Protrusion cues refer to local deformities where a 2D shape region protrudes from the main body resulting in a new branch segment in the 2D skeleton (see Fig. 3). On the other hand, [19] attempts to find n by minimizing the Akaike and Bayesian information criterion (AIC/BIC) with an iterative search over n and the parameters for each ellipse, incurring a large run-time overhead. Our contribution is two-fold: first, we propose a fast local fitting method that decomposes a 2D shape into candidate regions based on intuitions drawn from [2, 5, 10] and selects a set of ellipses to describe each region by minimizing a cost function. Second, we propose a greedy ellipse merging procedure that considers all ellipses resulting from the local ellipse fitting and minimizes the overall cost of the sets of ellipses while maximizing the shape coverage.

We compare our method with three other ellipse fitting methods proposed by [19] and [31]. We test each method on over 4000 2D shapes of different objects, sizes and orientations from four datasets: LEMS, MPEG-7, SiSHA, and PASCAL Horses. We measure the shape coverage of each method and their respective run-times. Our method obtains higher shape coverage scores than [31] and gives comparable results to the methods proposed by [19] while beating all methods by a large margin in terms of run-time (see Table 1). Example output from our approach is provided in Fig. 1, 2 and 6, which also illustrate that our proposed ellipses correspond to parts (e.g. arms, legs, etc.) of an object that are meaningful to human understanding. The output of our method can be used for higher level processing such as generating a 3D ellipsoid model of an object class (e.g. horses) as shown in Fig. 6.

[1] Computed as the norm of the residual image between the object mask and the union of ellipse images.

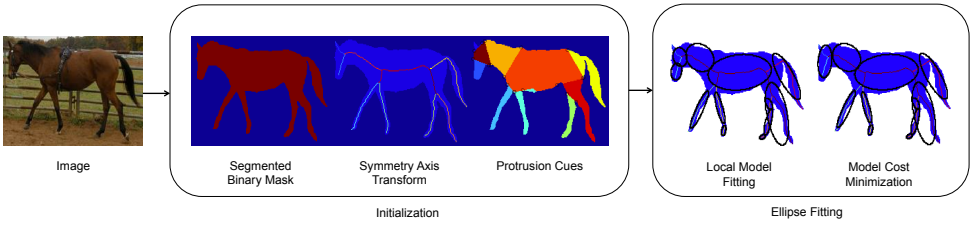

Figure 2: Overview of our approach. We initialize our approach via 2D skeleton and protrusion cues of the 2D object to produce a set of candidate regions for ellipse fitting. We apply our local ellipse fitting method and minimize the cost of the model for the final results.

2 Related Work

Our 2D shape modeling approach stems from principles of cognitive science and psychology, from which the studies describe the human perception of objects as compositions of parts [2] (e.g. a person has legs, arms, a torso, and a head). The minima rule, [5, 10], suggests that the perceived part boundaries of an object are generally located at points on a curve where there exists a negative curvature minima. Many works [13, 15, 22] follow the notion that convex regions determine visual parts. Moreover, [2] proposes that the object's semantic parts can be described by primitive geometric elements. Using these ideas, our 2D shape modeling approach automatically fits a set of ellipses that represents semantically meaningful regions.

Modeling 2D Objects with Ellipses. Modeling 2D objects with primitive shapes not only allows for the simplification of the data to benefit higher levels of processing, but also describes the geometric structure of the object, which can aid in the reconstruction of the object shape. While early works [4, 8, 17] utilized least-square methods for shape fitting, recent literature leans towards approximating the structure of an object through a combination of connected shape primitives. [31] uses an EM algorithm to propose ellipses based on a randomly generated set of initial guesses of where they are located and uses the Levenberg-Marquardt algorithm to solve for a least-square cost to adjust the ellipses to the 2D shape object. Our method for fitting ellipses to a 2D shape object is much simpler as we employ the 2D skeleton and protrusion cues to find candidate regions for the ellipse fitting step. We minimize Eqn. 2, which models the shape coverage and ellipse fitting costs and automatically determines the number of ellipses for the fit. [19] attempts to find the set of n-ellipses by minimizing the AIC/BIC criterion, which becomes a costly search over the large space of n and the ellipse parameters. We leverage the ideas of [2], [5] and [33], allowing us to approximate the n-ellipses for fitting the 2D shape object while considering local deformities.

2D Shape Decomposition. Many 2D shape decomposition methods leverage the minima rule [10] as a general criterion for how humans conceptually break objects into parts. The minima rule stems from the observation that humans generally perceive convex regions as important visual components to an object. [13, 14, 15, 22, 23] follows this intuition to measure the convexity of a region's boundaries to determine if the region should be classified as a distinct part. This important observation also leads to methods such as [29] and [33], which use the symmetry (medial) axes of a shape as a cue for convex parts. Moreover, [33] states that an important feature to determining whether a local convex region should be considered as a distinct part is the strength of protrusion given by its symmetry axes. A junction connecting three or more branches of the symmetry axes serves as a cue for multiple protrusions or convex regions extending from an object and a protruding region

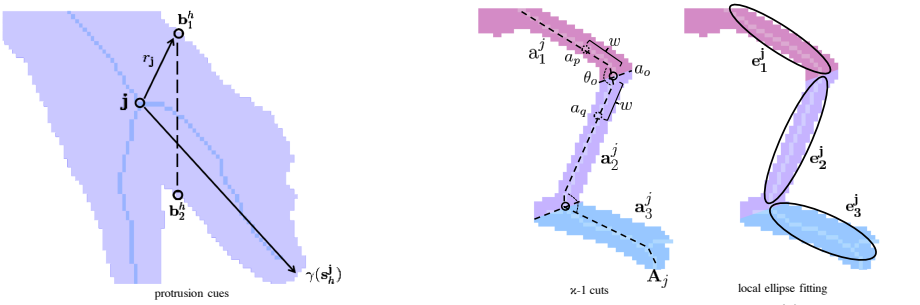

Figure 3: (a) To compute protrusion strength (Eqn. 1), we search outwardly from a junction $\mathbf{j}$ for the base of protrusion $\mathbf{b}_1^h$ and $\mathbf{b}_2^h$ at radius $r_\mathbf{j}$. $\gamma(\mathbf{s}_h^\mathbf{j})$ is the length of protrusion from $\mathbf{j}$. (b) We perform 2-cut over a region $\mathbf{R}^j$ with the symmetric axis $\mathbf{A}_j$ to generate 3 segments $\mathbf{a}_k^j$ for $k = \{1,2,3\}$ and their corresponding subregions $\mathbf{r}_k^j$ denoted by red, purple and blue. (c) We use least-square ellipse fitting to generate the local ellipse model $\mathbf{e}_k^j$ for the part region, $\mathbf{R}^j$.

will be considered a visual part if its protrusion strength exceeds some threshold.

3 Our Approach

Our goal is to model objects with ellipses that capture semantic meaning (i.e. correspond to distinct object parts). We do so by leveraging the Symmetry Axis Transform [3] and protrusion cues [33] to partition the foreground pixels into an initial set of candidate regions. We then compute a set of ellipses for each region via least square fitting [8]. Minimizing Eqn. 2, which attempts to maximize part coverage of the object shape while minimizing spurious ellipses, yields a set of ellipses modeling each region. A greedy cost minimization is then applied to the entire model to merge any suboptimal ellipses such that we maximize the coverage of the 2D shape. We provide an overview of our method in Fig. 2.

3.1 Initialization

Given a binary 2D object shape image $\mathbf{I} \in \{0,1\}^{M \times N}$, our goal is to estimate a set of ellipses to model the 2D shape where each ellipse holds semantically meaningful information about the 2D shape part that it is fitted over. We denote a 2D shape part as a set of pixels $\mathbf{P} \subset \mathbf{I}$ that defines a semantically meaningful region of the larger 2D shape. We first initialize the probable set of ellipses by decomposing the 2D shape into part regions using cues given by the skeleton $\mathbf{S}$ of the 2D shape, generated by applying Symmetry Axis Transform (SAT) [3] to the image $\mathbf{I}$. We use SAT as a shape descriptor because it is able to capture the structure of the 2D shape while providing a set of cues for decomposing the skeleton. Let $\mathbf{S}$ be a skeleton with H segments. We denote a segment $\mathbf{s}_h$ for $h \in \{1,2,\ldots,H\}$ in $\mathbf{S}$ as a set of singly connected pixels between a terminal pixel $\mathbf{t}$ and a junction $\mathbf{j} \in \mathbf{J}$ or between two junctions. The set of junctions $\mathbf{J}$ and segments $\mathbf{s}_h$ form a partition of the skeleton $\mathbf{S}$.

To generate each part region $\mathbf{P}_h \in \mathbf{P}$ corresponding to segment $\mathbf{s}_h$, we map each pixel in $\mathbf{I}$ to the nearest segment $\mathbf{s}_h$. Each segment in the skeleton implies that there is a protrusion of pixels in the shape and hence we can model such a set of pixels with an ellipse. However, the results of SAT are generally noisy. Therefore, rather than assuming that each of these segments should be a candidate ellipse, we employ protrusion cues from [33] to determine if a given segment and part region, $\mathbf{s}_h$ and $\mathbf{P}_h$, should be considered its own part or if it should

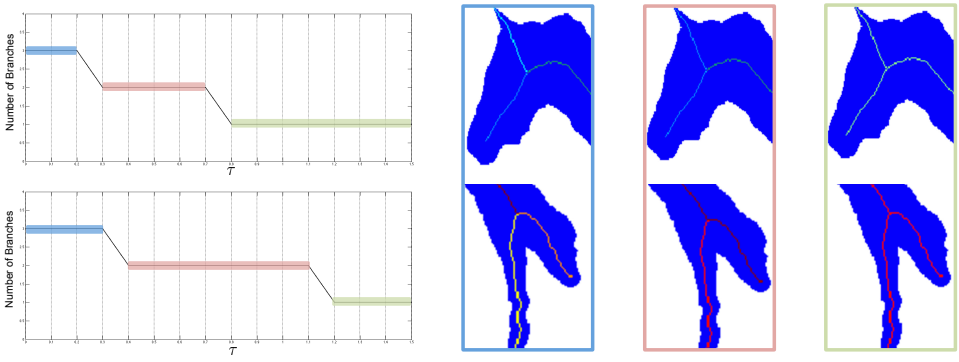

Figure 4: We show the affects of τ on determining the merging of the adjacent branches of the skeleton. Each color highlighting the horse heads and legs correspond to the number of branches given a set of τ values . Blue denotes three parts, red denotes two, and green one. We see that the number of branches are stable for a large range of τ values.

be merged with the part that $\mathbf{P}_h$ protrudes from (see Fig. 3).

When attempting to decompose an object, a protrusion from the main body of the object tends to highlight it as a potential part region. However, this protrusion may not necessarily correspond to a meaningful part (e.g. arm, leg, etc.) as each shape may contain small local deformities that are not semantically meaningful. Rather, it does serve as a good initial guess of where a candidate part region is located. For example, a hand may be decomposed into a palm and five fingers because the fingers protrude from the palm. However, we do not consider the small bumps on each finger as a semantic part because they do not protrude as significantly as the fingers do, which we determine by the protrusion strength. This is particularly the case when applying SAT to natural shapes. Because the shapes are imperfect, SAT will generate spurious branches. To mitigate this, we use protrusion cues to merge the spurious branches to major protrusions from the main object, making us robust to local deformations. To model this notion: for every junction $\mathbf{j} \in \mathbf{J}$ connecting the H segments $\mathbf{s}_h^{\mathbf{j}}$, we define the protrusion strength f_p for each segment $\mathbf{s}_h^{\mathbf{j}}$ as the ratio quantifying the amount a part region protrudes out from another. Fig. 3 shows an illustration of protrusion cues.

$$f_p = \frac{\gamma(\mathbf{s}_h^{\mathbf{j}}) - r_{\mathbf{j}}}{|\mathbf{b}_1^h - \mathbf{b}_2^h|} \tag{1}$$

where $\gamma(\mathbf{s}_h^{\mathbf{j}})$ is the length of $\mathbf{s}_h^{\mathbf{j}}$ from the junction to the peak of the protrusion, $r_{\mathbf{j}}$ is the radius from the junction $\mathbf{j}$ to the closest edge pixel of $\mathbf{I}$ and $\mathbf{b}_1^h$ and $\mathbf{b}_2^h$ denotes the left and right pixels sitting at the base of the protrusion. To identify $\mathbf{b}_1^h$ and $\mathbf{b}_2^h$, we take the set of edge pixels $\mathbf{p}$ within $r_{\mathbf{j}}$ distance of $\mathbf{j}$ and compute the sign of the cross product of vectors: $\text{sign}(\langle \mathbf{p}, \mathbf{j} \rangle \times \langle \mathbf{t}_h, \mathbf{j} \rangle)$ for a segment $\mathbf{s}_h^{\mathbf{j}}$ and its terminal point $\mathbf{t}_{\mathbf{h}}$. We separate the points whose sign is negative and positive and denote each sign as the left and right side of $\mathbf{s}_h^{\mathbf{j}}$. Given a junction $\mathbf{j}$ and the attached segments, if the protrusion strength f_p^h of a segment $\mathbf{s}_h^{\mathbf{j}}$ exceeds a threshold τ, then it will be considered distinct. Otherwise, the segment will be merged with the maximally protruding segment attached to $\mathbf{j}$ (see Fig. 4). This method is applied from the outside-in, beginning with segments that contain a terminal pixel $\mathbf{t}$. The result after applying the protrusion cues is the image mask containing the set of candidate regions, denoted as $\mathbf{R}$, whose non-maximal local protrusions are suppressed.

3.2 Local Model Selection

We estimate a set of ellipses $\mathbf{e}^j$ for each part region $\mathbf{R}^j \in \mathbf{R}$. Each $\mathbf{e}^j$ not only provides a tight fit over the region, but also models its structure (e.g. joints in a leg). We estimate the individual ellipses $\mathbf{e}^j_k \in \mathbf{e}^j$ by minimizing the cost function C (Eqn. 2), which describes the coverage of the model and its complexity. The first term (coverage) computes the mismatch between the areas of the proposed ellipses and the region. The latter term (model complexity) penalizes spurious ellipses that can be merged with other ellipses to accomplish a similar fit.

$$C(\mathbf{R}^j, \mathbf{e}^j) = ||\mathbf{R}^j - \bigcup_{k=0}^{\kappa} I(\mathbf{e}^j_k)||^2 + \kappa \frac{||\mathbf{R}^j||}{\eta(\mathbf{s}^\mathbf{j}) + 1} \tag{2}$$

where $I(\mathbf{e}^j_k)$ is the projection of an ellipse k into the image space for κ ellipses and $\eta(\mathbf{s}^\mathbf{j})$ is the number of branches in the segment $\mathbf{s}^\mathbf{j}$. Each $\mathbf{e}^j_k$ is fitted via a least square estimator. Increasing kappa improves fit but increases the complexity cost. Hence, Eqn. 2 allows us to find the best local fit while keeping the model complexity low.

To estimate the set of semantically meaningful ellipses $\mathbf{e}^j$ that models the joints and bends that may exist for a part region $\mathbf{R}^j$, we must have some understanding of the structure of $\mathbf{R}^j$ and whether or not this shape can be further decomposed into subregions. We borrow the intuition from the well-studied minimal-rule [2, 5], which proposed that the human perceived part boundaries of an object are generally located at points on a curve where there exists a negative minima. To model this intuition, we first apply SAT to the part region $\mathbf{R}^j$ to obtain a skeleton of the part. Then we choose the longest continuous segment through the major axis of the part region as the symmetric axis $\mathbf{A}_j$ of $\mathbf{R}^j$. For each point $a_o \in \mathbf{A}_j$, there exists the points on the symmetric axis, a_p and a_q, that are w pixels away on either side of a_o. The three points form the vectors, $\vec{a_{po}}$ and $\vec{a_{qo}}$, with which we compute the inner angle $\theta_o \in \theta_A$ using four-quadrant inverse tangent at every point a_o:

$$\theta_o = \tan^{-1}\left(\frac{\left\|\vec{a_{po}} \times \vec{a_{qo}}\right\|}{\vec{a_{po}} \cdot \vec{a_{qo}}}\right) \tag{3}$$

To produce the κ ellipses to model the part region $\mathbf{R}^j$, we perform $\kappa - 1$ cuts along the symmetric axis $\mathbf{A}_j$ (Fig. 3a). The value of κ may vary from each part region as the final κ will be selected by minimizing the fitting cost C (Eqn. 2) over all values of κ. For each point $a_o \in \mathbf{A}_j$, we compute the inner angle θ_o over a window of size $2w$ where the two points a_p and a_q defining the vectors $\vec{a_{po}}$ and $\vec{a_{qo}}$ are w pixels away from a_o. We perform non-minimal suppression over the set of θ_A within their respective windows and choose the top $\kappa - 1$ locations where the set of θ_A is smallest, implying the greatest change in the local gradient direction. The symmetric axis $\mathbf{A}_j$ is then cut at each location, producing a set of κ segments $\mathbf{a}^j \subset \mathbf{A}_j$. We then map each pixel in the part region $\mathbf{R}^j$ to the nearest segment $\mathbf{a}^j_k \subset \mathbf{a}^j$ to generate a set of subregions $\mathbf{r}^\mathbf{j} \subset \mathbf{R}^j$ where each subregion $\mathbf{r}^j_k \in \mathbf{r}^j$ correspond to each $\mathbf{a}^j_k$.

Given the set of subregions $\mathbf{r}^j$, we fit a candidate ellipse $\mathbf{e}^j_k$ using least-square over each subregion $\mathbf{r}^j_k$ to produce the set of candidate ellipses $\mathbf{e}^j$ to model the part region $\mathbf{R}^j$ (Fig. 3b). We begin estimating our set of candidate models starting from $\kappa = 1$ where we do not make a cut over the symmetric axis $\mathbf{A}_j$ and fit a single ellipse over the entire region $\mathbf{R}^j$. We then increase the number of cuts by one on each subsequent iteration. We compute the cost of the candidate ellipses $\mathbf{e}^j$ via Eqn. 2 on each iteration and denote the cost as ∞ if we are unable to fit an ellipse $\mathbf{e}^\mathbf{j}_\mathbf{k}$ over a subregion $\mathbf{r}^j_k$. This occurs when the estimated points of the ellipse

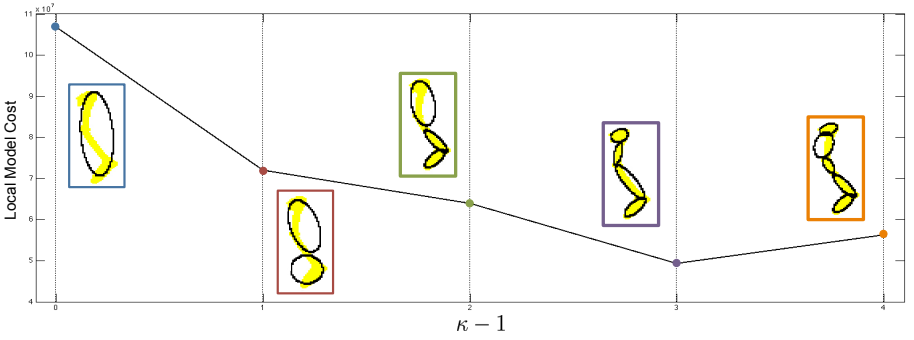

Figure 5: The results of local ellipse fitting with the cost of each model given the output of $\kappa - 1$ cuts. Given the local shape region, we generate $\kappa - 1$ cuts along the symmetric axis and fit κ ellipses to each sub-regions. The cost associated with each κ is given by Eqn. 2. The local model with the minimum cost is selected.

lie outside of the $N \times M$ space of the image I or when the least square fitting fails due to matrix inversion. When these cases happen, we terminate the $\kappa - 1$ cut procedure. The best set of ellipses $\hat{\mathbf{e}}^j$ modeling each of the local part regions $\mathbf{R}^j$ of the 2D shape are selected by minimizing the cost function C (Eqn. 2) over all proposed sets of candidate ellipses $\mathbf{e}^j$.

3.3 Model Cost Minimization

Upon completing the local ellipse fitting procedure, we take the union of the all sets of $\hat{\mathbf{e}}^j$ as the initial fitting of all ellipses to model the 2D shape image in $\mathbf{I}$. We denote the set union as $\mathbf{E}$ and each ellipse as $\mathbf{E}_u$ for $u = [1, U]$. We now find the set of best fit ellipses globally over the entire 2D shape image $\mathbf{I}$ by only merging adjacent ellipses that may be better fitted by a single ellipse. We propose a greedy algorithm that finds a local minima for the overall model cost by choosing a pair of adjacent ellipses to merge at each iteration; the process terminates when no benefits can be gained by merging any pair of adjacent ellipses (refer to the final step in Fig. 2, where the ellipses representing the horse's head are merged).

We first assign each pixel in the 2D shape to the closest ellipse to produce a region $\mathbf{Q}_u \in \mathbf{Q}$ where each $\mathbf{Q}_u$ corresponds to each $\mathbf{E}_u$. Using the regions $\mathbf{Q}$, we define the adjacency graph $\mathbf{G} \in \{0,1\}^{U \times U}$, where regions $\mathbf{Q}_u$ and $\mathbf{Q}_v$ are adjacent if $\mathbf{G}_{uv}$ is 1. For all pairs of adjacent regions, we compute the cost of replacing the pair of regions with a single new ellipse E_{uv} to fit over both of them with the following cost function:

$$D_{uv} = C(\mathbf{Q}_u, \mathbf{E}_u) + C(\mathbf{Q}_v, \mathbf{E}_v) - C(\mathbf{U}_{uv}, \mathbf{E}'_{uv}) \tag{4}$$

We choose the pair $\mathbf{E}_u$ and $\mathbf{E}_v$ to be replaced by $\mathbf{E}'_{uv}$ such that $\mathbf{E}'_{uv}$ maximizes the error reduction over the entire model:

$$(u, v) = \arg\max_{u,v}(D_{u,v}) \tag{5}$$

Then G is rebuilt with the new set of ellipses. We perform this greedy strategy over the entire model until the $\max(D_{u,v}) < 0$, in which case we obtain no gain from merging any of the adjacent ellipses. We denote the final set of ellipses as the output of our method.

4 Experimental Results

Four models were tested on the ellipse fitting task: our method, Augmentative Ellipse Fitting Algorithm (AEFA) and Decremental Ellipse Fitting Algorithm (DEFA) from [19] and the

	Intersection-Over-Union				Run-time (seconds)			
Method	LEMS	MPEG-7	SiSHA	Horses	LEMS	MPEG-7	SiSHA	Horses
Ours	0.9634	0.9597	0.9638	0.9507	3.88s	2.20s	2.19s	19.96s
DEFA	0.9623	0.9615	0.9629	0.9545	211.44s	255.87s	125.28s	2503.21s
AEFA	0.9711	0.9600	0.9654	0.9527	745.45s	1001.92s	528.36s	12088.02s
EMAR	0.9326	0.9219	0.9600	0.9470	120.57s	78.41s	63.23s	740.81s

Table 1: Intersection-Over-Union (IOU) results (left) and run-time (in seconds) results (right) for each method. Each method was tested on a single core. Our method provides comparable results to the competing methods while outperforming them by a large margin in run-time.

EM algorithm (EMAR) described by [31]. We ran each model on over 4000 shapes given by LEMS, MPEG-7, SiSHA, and PASCAL Horses. We allow each method to choose the optimal number of ellipses to describe the 2D shape object based on their own metrics for model selection. We measure the Intersection-Over-Union (IOU) of the output ellipses from each method and the input mask. The mean IOU is computed across each dataset to show the shape coverage of each method. The mean run-time of each method on a single core is measured in seconds. We show that our method performs comparably to competing methods while outperforming the competition in mean run-time.

For the experiments, we use an Intel processor with eight cores running at 2.99 GHz with 32 GB of physical memory. We would like to highlight that our method involves a number of inexpensive computations, which in fact allows us to run our model on a conventional laptop. In our experiments, we set w (Sec. 3.2) to 5 pixels and the protrusion threshold τ (Sec. 3.1) to 0.5. We used the implementation of AEFA, DEFA, and EMAR given by [19].

Examples of our output are provided in Fig. 1, 2 and 6. Results of our experiments are summarized in Table 1. We show that our method performs comparably to DEFA and AEFA (where the maximal change in mean IOU between our method and DEFA and AEFA is less than 0.01) while consistently outperforming EMAR [31] in mean IOU (Table 1 left). However, we see a significant difference in the run-times of each method (Table 1 right). While AEFA and DEFA outperforms EMAR in terms of mean IOU, EMAR experiences a 3-fold improvement in speed over DEFA and approximately14-fold over AEFA. Our method, however, compares well with both DEFA and AEFA in mean IOU, but instead we gain a 109-fold of mean run-time improvement over DEFA and 508-fold over AEFA. Our implementation runs on a single core, however, our local ellipse model fitting can be applied in parallel to part regions, which would yield an even greater speed up.

We note that DEFA, AEFA and our method use SAT [3] as a way of initializing each respective algorithm. Due to this similarity, each method shares an initial set of parameters for the ellipses and hence the comparable performance across the datasets. Despite this similarity, each method may differ in number of ellipses n and vary in model complexity due to their fitting criterion. In order to estimate n, DEFA and AEFA performs an iterative search while minimizing the AIC/BIC criterion. This results in slow run-times due to the large search space over n and the ellipse parameters. Our method estimates n via protrusion cues and our fitting criterion (Eqn. 2), which are computationally inexpensive. This makes our method a feasible candidate for a mid-level representation of objects in a larger image or video processing pipeline. In addition, our method produces ellipses that describe semantically meaningful structures (e.g. arms and legs) and captures joint locations of such structures while being robust to local deformities.

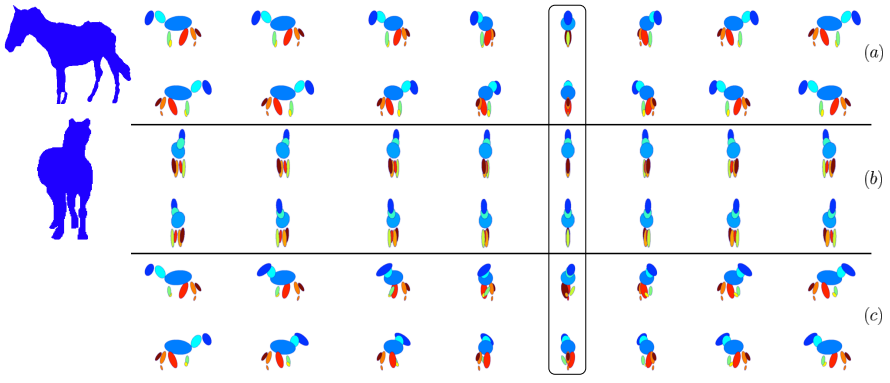

Figure 6: Pseudo-3D model built by applying our method to two images in PASCAL horses. (a) denotes the model of the profile view of a horse. (b) denotes the model the frontal view of another horse. (c) denotes a model generated by combining (a) and (b). We rotate the ellipses about the vertical axis by 22.5° increments. In (a) and (b), we have no depth information (we assume all ellipsoids lie on the same plane), and thus all parts align perfectly in the highlighted column (90° and 270° rotations). We match the ellipses to the semantic parts of the horses and establish correspondence between the two sets, allowing us to generate a pseudo-3D model. Despite the two images being taken from different horses at different viewpoints and poses, our method is able to register the two and render a richer model representing the horses's 3D structure (see separated legs in the highlighted column).

5 Application of our Model

One application of our 2D ellipse models is to construct a pseudo-3D horse model composed of geometric primitives (e.g. ellipsoids). We show a simple example of this in Fig. 6 by combining the ellipse models obtained from two viewpoints (side-profile and frontal view). The resulting model can act as a mid-level representation of the object, which can facilitate the detection and segmentation of horses from additional viewpoints. The process of generating it is computationally cheap and described below.

Let x, y, z represent a 3-coordinate system where x, y describes the coordinates in the 2D image and z is a pseudo-depth estimate. Given the output from our approach, we first assume that all of the ellipses lie on the same plane ($z = 0$) since we do not have any depth information. We then generate a pseudo-3D representation of our 2D ellipses model by performing perspective transformation on the ellipses and scaling the ellipses based on depth. For simplicity, we restrict our set of transformations to rotations about the vertical axis as the images in the dataset reflect only these poses as shown in Fig. 6. We transform our ellipses into ellipsoids using the short-axis, a, of each ellipses to model the "thickness" of each part. The ellipse projection proceeds as follows: We model each ellipsoid with two points, the center $\bar{X}$ (the pseudo-3D coordinate of $\bar{X} = [\bar{x}_1, \bar{x}_2, \bar{x}_3]$, initialized as $\bar{X} = [\bar{x}_1, \bar{x}_2, 0]$) and a point along the long axis $\bar{Y}$, and b. Next, we rotate the ellipse about the vertical axis by transforming $\bar{X}, \bar{Y}$ with the rotation matrix:

$$R(\phi) = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \tag{6}$$

We define the rotated coordinates as $\bar{X}^{\phi} = R(\phi)\bar{X}$. Projecting the points onto the image

plane by normalizing by their z coordinate allows us to recompute a and ϕ from $\bar{X}^{\phi}$ and $\bar{Y}^{\phi}$. In combination with b, we have all the ellipse parameters and can re-render the 2D ellipses from unseen angle ϕ, creating a pseudo-3D model of the object. We define a set of confidence scores $\bar{C} = [c_1, c_2, c_3]$ corresponding to each coordinate where $\bar{C}$ is initialized as $[1,1,0]$. We compute the confidence scores at each rotation $\bar{C}^{\phi}$ via $\bar{C}^{\phi} = R(\phi)\bar{C}$.

We perform this procedure for each set of ellipses. We then map each ellipses to the semantic parts of the horses (e.g. head, torso, front right leg) and establish correspondence between two sets of ellipses. Given that multiple ellipses may represent the same part, we choose the set with the greater number of ellipses as our method can locate joints via our $\kappa - 1$ cuts procedure (Sec. 3.2). Upon establishing correspondence, we combine the two models by taking the weighted linear combination of their respective $\bar{C}^{\phi}$ and $\bar{X}^{\phi}$ to generate the new pseudo-3D coordinates. As seen in Fig. 6c, we can estimate depth of the legs as shown by the spacing of the leg in the highlighted 90° and 270° rotated frontal image. With more samples, we can learn a 3D ellipsoid model using our approach as an intermediary step.

6 Discussion

Ellipse fitting becomes an important problem as it can serve as the next step in modeling segmented objects from semantic segmentation and boundary detection in images and videos. We introduce a technique for modeling 2D objects by fitting a set of ellipses over the shape. Our approach finds an initial set of candidate regions to fit the ellipses using a 2D skeleton with protrusion cues (Sec. 3.1). We further decompose the regions into sub-regions using the $\kappa - 1$ cuts approach (Sec. 3.2) and determine the set of ellipses by minimizing Eqn. 2. The cost is then minimized via Eqn. 4, 5 to produce the final model. We validate our approach on over 4000 2D shape images, illustrating comparable performance to recent work while taking a significantly lower computational toll. Our method is useful as an intermediary step for part-based methods such as constructing a mid-level representation of an object.

Our method does have room for improvement as the IOU scores show some margin of error. However, it is difficult to model every part of a shape with an ellipse. For example, multiple ellipses are needed to accurately fit the fine-grained corners of a region, but this increases the model complexity. We can extend our method to include other shape primitives (e.g. triangles) to improve the fit, however, this would also add to the complexity of the approach as we need to choose the optimal shape or shape combination for a region.

There exist two hyper-parameters w and τ that we need to tune. To automatically find these parameters, we can employ a multi-scale framework to produce models based on different window sizes or infer the protrusion threshold based on the skeleton of the 2D shape. Although this will increase our computational complexity, our approach will scale with parallelism since the local-ellipse fitting per region is computed independently. However, we have used the same parameters for all four datasets where the images varied in sizes and poses and we showed our method to be effective on each dataset.

Our results show that our method is an effective framework for modeling objects and that there are many avenues we can explore. Can we extend to general shape primitives? Our method has also proven to be able to locate ellipses that correspond to semantic parts of an object. Is it possible to extend our framework for semantic parsing? We plan to explore these directions given the promising results of our work.

Acknowledgements : This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) contract number D16PC00007 and NSF award CCF-1317376.

References

[1] Xianye Ben, Weixiao Meng, and Rui Yan. Dual-ellipse fitting approach for robust gait periodicity detection. *Neurocomputing*, 79:173–178, 2012.

[2] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[3] Harry Blum. Biological shape and visual science (part i). *Journal of theoretical Biology*, 38(2):205–287, 1973.

[4] Fred L Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9(1):56–71, 1979.

[5] Myron L Braunstein, Donald D Hoffman, and Asad Saidpour. Parts of visual objects: An experimental test of the minima rule. *Perception*, 18(6):817–826, 1989.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[7] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Transactions on Graphics (TOG)*, 34(6):195, 2015.

[8] Andrew Fitzgibbon, Maurizio Pilu, and Robert B Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5): 476–480, 1999.

[9] Nikos Grammalidis and Michael G Strintzis. Head detection and tracking by 2-d and 3-d ellipsoid fitting. In *Computer Graphics International, 2000. Proceedings*, pages 221–226. IEEE, 2000.

[10] Donald D Hoffman and Whitman A Richards. Parts of recognition. *Cognition*, 18(1): 65–96, 1984.

[11] Bogdan Kwolek. Stereovision-based head tracking using color and ellipse fitting in a particle filter. In *European Conference on Computer Vision*, pages 192–204. Springer, 2004.

[12] Nikolaos Kyriazis and Antonis Argyros. Scalable 3d tracking of multiple interacting objects. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3430–3437. IEEE, 2014.

[13] Longin Jan Latecki and Rolf Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441–454, 1999.

[14] Jyh-Ming Lien and Nancy M Amato. Approximate convex decomposition of polygons. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 17–26. ACM, 2004.

[15] Hairong Liu, Wenyu Liu, and Longin Jan Latecki. Convex shape decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 97–104. IEEE, 2010.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[17] Gabor Lukács, Ralph Martin, and Dave Marshall. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *Computer VisionâĂŤECCV'98*, pages 671–686. Springer, 1998.

[18] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 743–751, 2016.

[19] Costas Panagiotakis and Antonis Argyros. Parameter-free modelling of 2d shapes with ellipses. *Pattern Recognition*, 2015.

[20] Costas Panagiotakis, Emmanuel Ramasso, Georgios Tziritas, Michèle Rombaut, and Denis Pellerin. Shape-based individual/group detection for sport videos categorization. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(06):1187–1213, 2008.

[21] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 376–383. IEEE, 2014.

[22] Zhou Ren, Junsong Yuan, Chunyuan Li, and Wenyu Liu. Minimum near-convex decomposition for robust shape representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 303–310. IEEE, 2011.

[23] Zhou Ren, Junsong Yuan, and Wenyu Liu. Minimum near-convex shape decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(10):2546–2552, 2013.

[24] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015.

[25] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Xiang Bai, and Alan Yuille. Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images. *arXiv preprint arXiv:1609.03659*, 2016.

[26] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4268–4276. IEEE, 2015.

[27] Alexander Toshev, Ben Taskar, and Kostas Daniilidis. Shape-based object detection via boundary structure segmentation. *International journal of computer vision*, 99(2): 123–146, 2012.

[28] Nhon H Trinh and Benjamin B Kimia. Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *International Journal of Computer Vision*, 94(2):215–240, 2011.

[29] Chun Wang and Zhongyuan Lai. Shape decomposition and classification by searching optimal part pruning sequence. *Pattern Recognition*, 2016.

[30] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[31] Da Xu, Richard Yi, and Michael Kemp. Fitting multiple connected ellipses to an image silhouette hierarchically. *Image Processing, IEEE Transactions on*, 19(7):1673–1682, 2010.

[32] Lei Xu and Erkki Oja. Randomized hough transform (rht): basic mechanisms, algorithms, and computational complexities. *CVGIP: Image understanding*, 57(2):131–154, 1993.

[33] Jingting Zeng, Rolf Lakaemper, Xingwei Yang, and Xin Li. 2d shape decomposition based on combined skeleton-boundary features. In *Advances in Visual Computing*, pages 682–691. Springer, 2008.