Interactive Sketch-Driven Image Synthesis

Daniyar Turmukhambetov [1], Neill D. F. Campbell [1,2], Dan B Goldman [3], Jan Kautz [1,4]
[1] University College London, United Kingdom [2] University of Bath, United Kingdom
[3] Adobe Research, WA, USA [4] NVIDIA Research, MA, USA

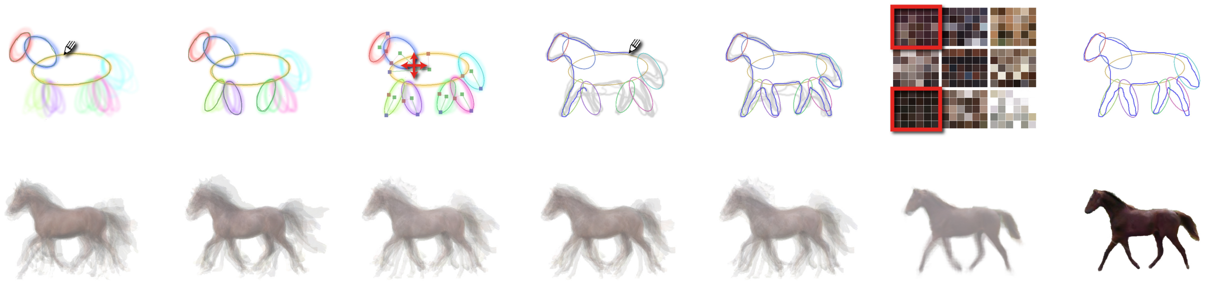


Figure 1: *Our interactive system guides a user to specify pose and appearance using sketching, in order to synthesize novel images from a labeled collection of training images. The user first sketches elliptical "masses" (left), then contours (center), mimicking a traditional sketching workflow. Once the pose is specified, the artist can constrain the appearance and render a novel image (right). Top row: user sketch input and feedback guidelines; Bottom row: rendered previews.*

Abstract

We present an interactive system for composing realistic images of an object under arbitrary pose and appearance specified by sketching. Our system draws inspiration from a traditional illustration workflow: The user first sketches rough "masses" of the object, as ellipses, to define an initial abstract pose that can then be refined with more detailed contours as desired. The system is made robust to partial or inaccurate sketches using a reduced-dimensionality model of pose space learnt from a labelled collection of photos. Throughout the composition process, interactive visual feedback is provided to guide the user. Finally, the user's partial or complete sketch, complemented with appearance requirements, is used to constrain the automatic synthesis of a novel, high-quality, realistic image.

Keywords: sketching, drawing, image based rendering, image synthesis

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Interaction techniques—Applications

1. Introduction

Suppose you would like to create a realistic image of an animal – a horse, for example. You can imagine the horse's pose and appearance in your mind's eye, but how can you translate that vision to an image? Painting realistic images with correct proportion, pose and color requires training and talent that most of us lack. Thanks to Google and other search providers, it has become easier for novices to search the internet for images of a given object category using keyword search. But finding examples in a particular pose or relationship to the camera requires searching through pages and pages of search results. Furthermore, it's possible that no one image matches what you originally imagined: perhaps one image almost has the correct pose, another has details you like, and a third has the color you had in mind. Combining and modifying these images to produce one matching your goal is very involved, even with state-of-the-art image editing software.

In this paper, we present an end-to-end system that enables users to interactively sketch and synthesize novel images, given a database of labelled images. When designing a system for controlling image synthesis, a critical challenge is defining user interactions that are meaningful to a human but also express appropriate constraints on the synthesis. To address this challenge, we draw inspiration from the strategies employed by figure drawing artists: it is common for such artists to begin by sketching the rough body proportions using overlapping ellipses or other simple abstract shapes, often called *masses*. Once the masses are sketched as desired, the artist can go on to add contours, shading and finer details of the figure. While this serves as a useful starting point, it is not sufficient to serve non-expert artists: We cannot expect a casual user to produce a sketch that resembles a realistic image as input to a synthesis engine without assistance. Therefore the system must guide the user in an exploration of the synthesis space, rather than simply optimizing a set of pre-defined constraints.

More concretely, we propose that a sketch-driven image synthesis system must meet four intertwined design principles: First, it must be *responsive*, providing results rapidly enough for a user to iteratively refine her mental concept. Second, it must be *exploratory*, guiding the user through the space of possible syntheses with meaningful feedback. Third, it must be *robust* to missing data, providing meaningful feedback in the face of incompletely-specified constraints. And fourth, the interactions should be as *fluid* as possible – preferring sketch gestures over menu or label selections.

These principles have driven all of the design choices for our system: In the initial "mass" phase we support freehand sketching of new masses. Our system supports traditional "overdraw" to adjust these masses after their initial placement, but we also provide direct manipulation of mass ellipses. A preview is shown at all phases, which increases in specificity and concreteness as the constraints are refined. The preview is updated after each sketch gesture, rather than waiting until the sketch is complete. In early stages this takes the form of blended images from the database, giving a visual representation of the local space that meets the constraints specified thus far.

We also provide feedback in the form of shadowed lines underneath the user's sketch. These shadows suggest, first, suitable ellipse locations and, second, valid object contours. The recommendations are derived from a probabilistic model of the labeled body-parts in an image database and help an untrained user to draw masses and contours in appropriate layouts for each object. We adapt existing techniques to synthesize a final image that is consistent with the user's sketch.

The masses in our system are used for two purposes: they serve as a proxy for specifying pose and they are used to guide the final synthesis stage.

Artists use a variety of geometric primitives as masses: The human pelvis is often represented using a cuboid, and arms with cylinders. Human faces can be represented using ellipsoids sectioned at various ratios along their axes (similarly to reference lines in [DPH10]). Although our system could be augmented to support numerous primitives, we focused our initial efforts on a single one – the ellipsoid. Although it is not commonly used in sketching humans, this primitive is flexible enough to sketch a variety of walking and flying animals, and thus our datasets span such animals. To our knowledge, ours is the first image synthesis system that interleaves sketched constraints and preview synthesis. We argue that this interaction modality facilitates joint human-computer exploration of a space of synthesized images, and is thus the foremost contribution of our work.

In addition, we apply machine learning techniques to learn a low-dimensional manifold from the data that models the joint configuration of masses and the contour shape of objects, a highly complex relationship that could not be specified using a heuristic approach. We bring together appropriate representations for the masses (Stokes parameters) and the contours (elliptical Fourier coefficients) to ensure that the manifold interpolations are plausible even when using relatively few input samples, unlike ShadowDraw which requires a dense sampling of each sketch configuration space. For the final synthesis stage we adapt the Image Melding algorithm [DSB*12] by using additional guiding layers corresponding to each distinct body part and the silhouette.

2. Related Work

Some existing systems, such as Sketch2Photo [CCT*09] and Johnson *et al*. [JBS*06], have merged retrieval and synthesis into unified systems. Both allow a user to sketch a query combining text, images, and/or outlines; retrieve matches from the internet or a local database; and *compose* a new image using the retrieved elements. Goldberg *et al*. [GCZ*12] use the Sketch2Photo framework to allow object-level manipulation in images using online images queried by the user's keywords and segmentations. They propose novel deformation and alignment techniques to achieve high-quality results. However, neither method is targeted for interactive use: Johnson *et al*. report 15 to 45 seconds per composition, Tao *et al*. report 15 minutes for each object retrieval and another 5 minutes for composition; and Goldberg *et al*. report 10 minites for object retrieval. We posit that a more interactive system is critically important for creative processes, allowing users to quickly explore the space of possible outcomes.

Photo clip-art [LHE*07] and Photosketcher [ERH*11] are two other systems that are designed for interactivity. Photosketcher, in particular, uses sketches as input. Similarly to Photosketcher, Sketch2Scene [XCF*13] uses user sketches to retrieve and place 3D models to assist the user in the 3D scene modeling from a dataset of 3D models. However, like the other methods, these systems *compose* the final output using only one image or 3D model per object, rather than synthesizing new objects using combinations of retrieved images. Furthermore, none of these previous approaches provide a mechanism for detailed pose specification.

Recently the PoseShop [CTM*13] system proposed using online image search to construct a segmented human image database with pose and semantic action descriptions. PoseShop queries the database with a sketch or skeleton, allowing a composition of personalized images and multi-frame comic strips by swapping the head and clothes to the user's specifications. This system does compose novel images, but doesn't provide interpolation between poses and uses only one database image for each output.

Other recent works blend elements from a collection of related images. For example, Mohammed *et al.* [MPK09] learn a global parameterized model of frontal face images, and constrain a patch-based texture synthesis algorithm using a sample from this model. Risser *et al.* [RHDG10] demonstrated a hierarchical pixel-based texture synthesis algorithm that generates novel image hybrids by jittering exemplar coordinates instead of spatial coordinates, in order to preserve structures. However, neither of these systems supports pose variation or provides direct user control of the synthesized result. The PatchNet system combines multiple input images for each output, taking into consideration contextual relations between parts [HZW*13]. However, it was only demonstrated for scene composition rather than object posing, as it does not incorporate an explicit model of pose. Recently Darabi *et al.* [DSB*12] demonstrated a patch-based system called "image melding" to smoothly interpolate textures and colors across images. Our synthesis engine builds on the image melding framework with additional control channels, as in image analogies [HJO*01].

In the domain of drawing assistance, Lee *et al.* recently proposed a system that allows freeform drawing of objects [LZC11]. As the user adds strokes, the system interactively provides shadow-like hints of where the next stroke should be. The system doesn't have any prior knowledge about the object that the user is drawing, so the shadows are constructed by blending relevant edge-maps from a large database queried using local edge patch descriptors. Sketch-Sketch Revolution [FGF11] allows novice users to learn to replicate strokes of an expert artist by following guidance and feedback of the system in a step-by-step tutorial, which was previously created by an expert artist. The iCanDraw? [DPH10] system assists users by generating *corrective* feedback extracted from a reference image. Similarly, The Drawing Assistant [IBT13] automatically extracts *"block in"* visual guides from a single reference image and provides corrective feedback to the user. Sketches produced with the help of these systems have more realistic proportions than unassisted drawings, but the outputs are only contour sketches, not realistic images. Inspired by these systems, we strive to utilize sketch inputs to produce plausible realistic image outputs, restricting our attention to the case in which the class of object is known in advance.

3. Sketch Interaction

One traditional sketching method is to first draw the outline of the subject using primitive shapes likes ellipses, circles, squares, etc. [Bla94, ES07, Ful11, Dra]. At this stage only the gross relationships of body parts are specified, and the simplicity of the shapes makes it possible to adjust and iterate quickly. It is important that these shapes are very basic, allowing the artist to define the correct proportions between parts of the object without focusing attention on small scale details. Once the rough outline of the object is set, the artist can add finer details. Beyond this point, the masses act as an anchor for the drawing, so that the artist can focus on local details without breaking proportion or symmetry. An example of this approach is given in Figure 2, showing an artist using masses to sketch a horse and a pigeon. Our goal is to mimic this approach by providing visual feedback that can guide the user in adjusting masses and defining strokes.

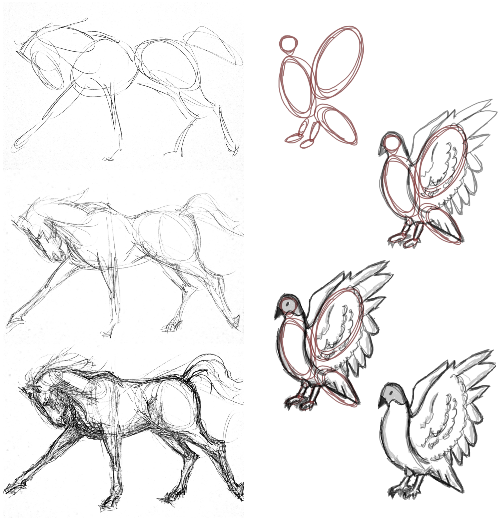


Figure 2: *Sketching a pigeon and a horse by hand using masses. The artist starts by drawing in the masses for the body parts in the correct proportions before continuing to fill in the contours and then any final details, such as shading.*

We argue that the use of elliptical masses is a more effective tool for specifying gross shape than either contours or "skeletons." Contours contain both small and large scale detail, and it is typically difficult for a novice user to focus on both scales at the same time as they trace an outline. Skeletons or "bones" are an appealing alternative due to their ubiquitous use in 3D computer graphics. However, whereas a line can express only (2D) length and angle of a body part, an ellipse can also express its apparent thickness.

Perhaps more importantly, novice users are not necessarily good at evaluating the proper location of "bones" within a figure. For example, in Figure 3, where would you draw a straight line specifying the location of the neck (cervical vertebrae)? Most people without veterinary training will be surprised to see that when a horse's head is raised, the neck vertebrae of a horse are closer to the front of the neck at its base, but closer to the back of the neck at its apex. In contrast, elliptical masses require no knowledge of internal anatomy, and they form a visual guide for the subsequent stage of contour sketching, since the contours often follow close to the mass edges.

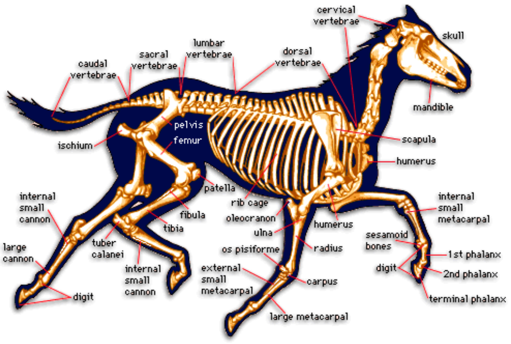


Figure 3: *A cross-section of a horse showing that the location of bones is unintuitive: Many people are surprised that the neck vertebrae are closer to the front of the neck when a horse's head is raised. (By courtesy of Encyclopaedia Britannica, Inc., copyright 1998; used with permission.)*

3.1. User Interaction

Our system's window shows two panels at all times: On the left, the *sketch* panel shows the user's sketched strokes, as well as semi-transparent guidelines suggesting possible stroke locations. On the right, the *preview* panel visualizes the space of possible output images, given the sketch progress thus far. Since the outcomes are increasingly constrained throughout the sketching process, the contents of both panes are constructed differently at various stages. The entire workflow is given in detail below, with descriptions of the panel contents seen at each stage:

Initial State The user is first shown an abstract overview of the range of pose and appearance of the target object. The sketch panel shows guidelines for the elliptical masses, indicating this is the first step in the sketching process. Since the range of masses can overlap a lot, the guidelines for each part are shown using a different color to improve visibility and comprehension. They are slightly blurry, to emphasize that they are only loose constraints: The user can sketch anywhere, but results are best when the sketches are close to the range of real object variation. The preview panel is empty, since the output is totally unconstrained at this stage.

Drawing Ellipses The user paints strokes in the sketch panel (in grey). After each stroke, the sketch panel shows ellipses (with the color of the estimated part) fitted to the strokes. The guidelines are updated to show plausible mass configurations similar to the user's sketched masses. The preview panel shows corresponding nearest-neighbour images, blended together using simple averaging, giving a ghosted view of possible outcomes (we call this "Fast NN Preview").

Mass Adjustment The user can adjust existing masses in two ways: Those familiar with traditional sketching may prefer "oversketching," simply drawing over the previous strokes to replace them. However, novice artists may prefer to adjust the mass ellipses using an object-oriented approach. For these users we provide an adjustment mode in which the ellipses can be directly manipulated: Colored handles appear on the major and minor axes of each ellipse in the sketch view. Dragging the center handle translates the corresponding ellipse, and dragging the axis handles rotates and/or scales it about the center. The preview panel is constructed in the same way as the previous paragraph.

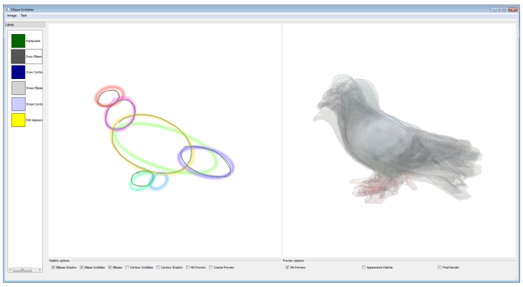
Drawing Contours In this mode, the sketch panel shows faded contours of real images similar to the user's sketch, much like ShadowDraw [LZC11], but interpolated using our manifold model.

Editing Appearance The previews described in previous paragraph blend together multiple input exemplars, and can thus obscure specific appearance details such as color and lighting. To address this, we also provide an appearance selection mode in which the preview window switches to a grid of color palettes computed from database exemplar that are near-matches to the current sketch. The user may click on one of these palettes to constrain the output appearance, and then return to sketching. The preview panel now shows a fast low-fidelity synthesis result aligning the images to match the sketched contour.

Final Synthesis When the artist is satisfied with the constraints and preview render, she can request a final rendering, which may take 3-4 minutes depending on the resolution of the images in the dataset. We use the user specified contour and the ellipses to guide the synthesis process, as this additional information helps the system deform the images of the dataset before blending them together.

Although we present the steps above in their logical sequence, the system does not require a strict linear progression through these stages: The modes can be revisited in any desired order. The artist may choose to constrain color before pose, or return to mass adjustment after drawing part or all of the object contour. Furthermore, the user can select appropriate visual feedback for each of the interactions.

Reflecting back on the four design principles proposed in Section 1: Our system is *responsive*, as both panels are updated after every user stroke. It is *exploratory*, because it attempts to illustrate at each stage the span of plausible outcomes given the current sketch. It is *robust*, by virtue of providing this feedback after a single user stroke, or hundreds. And it is *fluid*, utilizing hand-drawn strokes wherever possible; discrete menu or tool selections are required at only a few moments in a typical interaction. Figure 1 illustrates some stages from our workflow, and Figure 4 shows our system GUI running on the pigeon dataset. Please see supplementary materials for further examples.



Visibility options

☑ Ellipses Shadow ☑ Ellipse Scribbles ☑ Ellipses □ Contour Scribbles □ Contour Shadow □ NN Preview □ Coarse Preview

Preview options

☑ NN Preview □ Appearance Palette □ Final Render

Figure 4: *Our system's interface. The left panel is the user's drawing canvas, where the shadow feedback is shown. The right panel shows the fast nearest-neighbor preview.*

4. Implementation

In this section we provide details of the technical approach taken to provide the interactive workflow of the previous section. The most significant challenge is to provide a joint model of object pose, contour, and appearance space that a user can explore continuously and freely, within the constraints of plausible image synthesis. In addition, the system design is made more challenging by the requirement that everything must run at interactive rates, with the exception of the final image synthesis.

We propose an image melding [DSB*12] approach to synthesize novel images of a particular object: Our system combines multiple images of similar objects under similar poses to produce a final image in a specific pose. To allow for a wide range of poses and appearances of a particular object, we require a database of images containing differing poses and appearances. The prevalence of online image search and image libraries renders such an image database straightforward to obtain.

Once we have a collection of images to use for synthesis, we must consider how to address the fundamental requirements of our interactive workflow:

1. How do we identify reasonable configurations of masses to guide the user when specifying pose?
2. How do we identify feasible object contours given the pose of the object that will allow for accurate synthesis?
3. How can we inform the user of the possible variation in appearances of the object?
4. How do we select appropriate images to use to synthesize this specific pose and appearance?

We adopt a single methodology to deal with all of these questions; we make use of machine learning approaches to model the joint relationships of mass pose, object contour, and the training images (which encode object appearance) in a unified probabilistic framework. More specifically, we optimize, within the high dimensional joint space of mass poses and contours, a low dimensional manifold that contains all of the database images.

Using appropriate parameterizations, we can move continuously within the manifold, and smoothly interpolate the masses and contours between training images in order to generate valid novel poses. A location on the manifold identifies a specific pose, and the nearest neighboring database images in the manifold are good candidates for image synthesis. In addition to answering the above questions, the probabilistic nature of the model also allows us to handle image synthesis in the presence of incomplete information, *i.e.* missing masses or contours, affording our system a degree of robustness.

In order to train the model, we have to provide labeled data to associate the poses of each mass and the contours with each image in our database. This consists of segmenting each image into a set of body parts. From this segmentation we may fit a set of ellipses (as the object masses) and find the contour of the object.

Given this framework for feedback and synthesis, we must also consider the user input. User scribbles are interpreted as editable masses, allocated to specific body-parts, and the contour and appearance constraints must be specified. We now visit each specific component in further detail.

4.1. Training Data

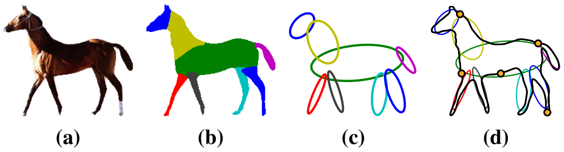

(a) **(b)** **(c)** **(d)**

Figure 5: *Example of training image segmentation. (a) The input image is segmented from the background then (b) split into its constituent parts to allow (c) ellipse fitting to represent the masses. (d) The contour of the complete silhouette and the alignment key points are then automatically extracted.*

We require a collection of images (dataset), containing a single class of objects, where the object of interest is segmented from the background. For each object, we define a model describing how the object is divided into masses. For example, a horse can consist of head, neck, torso, tail, left and right forelegs and hind legs.

We labeled each image by assigning each pixel to the relevant part of the object model. We then extract masses by fitting ellipses to the boundaries of each of the labeled parts. We chose ellipses to represent masses because they are popular among artists [Dra, Bla94], it is possible to fit them to curves efficiently [FPF99], and their shape is general enough to approximate many body parts. We note that any other shape with low-dimensional parameterization may be readily substituted in our system.

4.2. Joint Manifold

To produce good synthesis results, we must ensure that the training images and the user-specified ellipses and contours are compatible. For example, the head of a horse cannot be placed on the far end of its tail. To achieve this, we provide feedback to users during sketching. This requires a statistical model of the joint space of ellipses and contours covered by our training data. Such a model can estimate the likelihood of a particular arrangement of masses and contours; this is then used as a measure of how readily such a configuration may be reproduced from the training data.

We require that the model be sufficiently powerful to represent the complex interactions between the ellipses and the contour, a multi-modal distribution, and also allow for fast inference queries to be performed at interactive rates. We achieve both of these goals by representing the contours with elliptical Fourier coefficients [KG82] and modeling the joint manifold of the ellipses and contours using a Gaussian Process Latent Variable Model (GP-LVM) [Law05]. We now discuss each of these components in further detail.

Representation In order to interpolate smoothly between contours in different training images our system needs a continuous representation of the contour. However, the silhouette contours obtained from the segmented training images are not registered to one another with a dense correspondence. In any case, silhouettes from different viewpoints cannot be placed in meaningful correspondence; for example, the front legs of a horse may appear separately or on top of one-another. Inspired by the work of Prisacariu and Reid [PR11], we represent closed contours using elliptical Fourier coefficients [KG82], which can smoothly interpolate between the silhouettes of objects such as people, cars, and animals. Whereas Prisacariu and Reid used these silhouettes as a shape prior for segmentation and tracking, we will use it as a shape prior for image synthesis.

Fourier contour representations must be phase-aligned (*i.e.* a common starting point and parameterization) to achieve good interpolation [PR11]. We achieved high quality interpolation by aligning a series of key points distributed over the length of the contour. This corresponds to resampling the contour such that there are a fixed number of samples between each key point. We compute a sparse set of key points on the contour using the labeled parts in an automatic fashion; for example, a point that lies on the leg and is farthest from the torso, and a point on the torso that is closest to the tail. These key points were chosen to be empirically consistent between different poses and mass configurations. For the horse dataset, we use five key points as shown in Figure 5(d).

The general parametric form of an ellipse is expressed by 5 values $[x_c, y_c, a, b, \phi]$: the x-axis and y-axis coordinates of the center of the ellipse, the length of the major and minor axii and the angle between the x-axis and the major axis, respectively. Here, ϕ is in the range $[0, 2\pi]$. However, we require a smooth representation for the set of ellipses for each training image. We achieve this using Stokes parameters [McM54] that are defined as:

$$[x_c, y_c, a^2+b^2, (a^2-b^2)\cos(2\phi), (a^2-b^2)\sin(2\phi)]$$

Manifold Given the continuous representation, it is possible to interpolate between similar training images to generate new pose configurations of ellipses and contours. However, we cannot, in general, interpolate linearly in the space of Stokes and Fourier coefficients. Instead we find a low dimensional manifold in the joint space of ellipses and contours that contains the training image configurations, using a Gaussian process latent variable model [Law05]. In cases where body parts are occluded we will be missing some ellipses; We employ the method of Navaratnam *et al.* [NFC07] to train a GP-LVM joint manifold model with missing data.

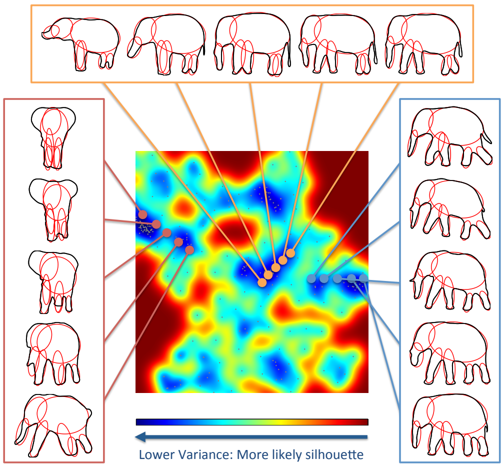


Figure 6: *A 2D joint manifold of ellipses and contours learnt for the elephant images. Each point represents a configuration in pose space, and the color indicates the variance of the embedding in the latent space. Regions with a low variance are higher probability in the pose space. The location of the original training images are shown as grey dots.*

The probabilistic nature of the GP-LVM model allows us to interpret the variance of the embedding in the low dimensional latent space as the likelihood of a pose given the training data. Figure 6 shows an example of the manifold learnt for a set of images of elephants. The coloring of the manifold shows the variance of the ellipses and contours that would be estimated at that point. Thus areas with a low variance (shown as blue in Figure 6) are more likely to produce good results under image melding from the local neighboring training images (shown as grey dots on the manifold).

4.3. Ellipse and Silhouette Queries

We now provide details of how to provide the shadowed ellipses and contours used during the interactive sketching process. We first consider how to identify plausible locations for the remaining ellipses given that the user has already drawn one or more of the masses. We then consider how to identify reasonable object contours given a set of masses and partially sketched contour fragments. Whilst both these approaches are used to provide visual guidance to the user sketching process, they can also be used to fill in incomplete data for the final synthesis (*e.g.* if the user has missed some

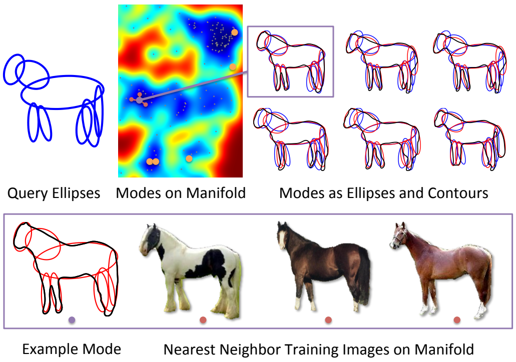

Query Ellipses
Modes on Manifold
Modes as Ellipses and Contours
Example Mode
Nearest Neighbor Training Images on Manifold

Figure 7: *Upper left: a set of user-specified ellipses (shown in blue) is used for a search over pose space (heat map, upper middle). Upper right: the modes of the distribution are shown in red over the original user specification in blue. Bottom: we show the three training images that are closest in the latent manifold space to the mode marked as a purple dot.*

of the masses or drawn an incomplete contour). In addition to the following descriptions, we provide some further details in the supplementary material.

Ellipse Manifold Queries We define a cost function based on the difference between the query ellipses – which may be partially specified – and a sampled set of ellipses from the latent space; we use the L-1 norm in the Stokes parameter space. Since we cannot evaluate analytic gradients we must perform this optimization using point estimates from the latent space (in a similar fashion to Navaratnam *et al.* [NFC07]). Because the function is multi-modal we start from a set of initialization points that span the latent space. This optimization can be performed rapidly since the dimensionality of the latent space is so low and the ellipse cost function is very cheap to compute.

Figure 7 shows an example set of query ellipses and the six most probable modes, both on the manifold and as the ellipses and contours. We also demonstrate that the training images with locations in the manifold closest to the mode have similar ellipse and contour layouts and are thus suitable to be used as source images in the synthesis stage.

Contour Manifold Queries Figure 7 demonstrates the multi-modal nature of the distribution of the contours with respect to the ellipses. In order to specify a particular silhouette, we allow the user to sketch parts of the contour. Just as we did above for an ellipse query, we define a distance function between the contour of a point on the manifold and user sketches: we use the chamfer distance [BTBW77] between the user sketch and the contour under a truncated-quadratic cost function. Since the cost function is truncated, the query results should be robust against incomplete and outlier sketches that the user may draw.

This defines the similarity between the user sketches and a point on the manifold. We then perform a set of optimizations (as described above for ellipses) to find the modes in the manifold. Since the contour function is more expensive to compute, we accelerate the search by using the modes of the ellipse query as the initializations. These queries typically take around half a second.

4.4. Sketching Masses and Contours

In the first stage of sketch interaction, we ask the user to scribble some or all of the masses. The ellipses that represent masses can be manipulated by dragging control points, but in order to make the user interface more intuitive, we also allow freeform drawing of masses, and fit the freeform strokes to a set of parameterized ellipses.

The user can sketch ellipses using one or many strokes, specifying ellipses either partially or completely. To solve this problem the system assigns each stroke to one ellipse, and each ellipse is fitted to all of its assigned strokes. As a user inputs a new stroke, the system computes the cost for each of the ellipses of the current set of ellipses:

$$\mathrm{cost}(i) = \frac{1}{||\{S^\star \cup S_{\varepsilon_i}\}||} \sum_{S_k \in \{S^\star \cup S_{\varepsilon_i}\}} \sum_{p \in S_k} \mathrm{dist}\left(p, \varepsilon_i^\star\right), \qquad (1)$$

where $S^\star$ is the new stroke, S_k is the k-th stroke of the user, S_{ε_i} is the set of strokes assigned to ellipse i, p is a point of the stroke S_k, $\varepsilon_i^\star$ is the ellipse that was fitted to $\{S^\star \cup S_{\varepsilon_i}\}$ and $\mathrm{dist}\left(p, \varepsilon_i^\star\right)$ is the distance from point p to the ellipse $\varepsilon_i^\star$.

This cost computes the average of the distances between the strokes that were assigned to the ellipse and the corresponding fitted ellipse. If the new ellipse stroke doesn't fit any of the previous fitted ellipses (the average distance is more than 40 pixels for each of the ellipses of the current set of fitted ellipses), the system creates a new ellipse. Otherwise, the new stroke is assigned to the best matching ellipse from the current set of fitted ellipses. This approach also allows the user to erase incorrect strokes and change ellipses by drawing over the top of existing ellipses. As long as the "overdrawn" strokes are nearby, the ellipse will fit all of the assigned strokes.

Having obtained a set of ellipses, we need to identify the corresponding body part label for each ellipse. This is performed automatically in two steps. First, we find the set of ellipses from the images of the dataset that are closest to the user's ellipses, by minimizing

$$i^* = \arg\min_i \sum_k \min_{j \in \mathcal{P}} \sum_{p \in \varepsilon_k} \mathrm{dist}\left(p, \varepsilon_{i,j}\right), \qquad (2)$$

where $\varepsilon_{i,j}$ is the ellipse fitted into the body part j of the labeled image i, and $\mathcal{P}$ is the set of body-parts, p is a point of ellipse ε_k, ε_k is the k-th ellipse fitted to the user's scribbles, and $\mathrm{dist}\left(p, \varepsilon_{i,j}\right)$ is the distance from point p to the ellipse $\varepsilon_{i,j}$.

We then obtain the part label for each of the user's ellipses:

$$j^* = \arg\min_j \sum_{p \in \varepsilon_k} \mathrm{dist}\left(p, \varepsilon_{i^*,j}\right). \qquad (3)$$

Once we assign a label j^* to ellipse ε_k, we remove j^* from the set of possible labels to ensure a unique assignment. We allow the user to override the assigned labels if desired. The operations above can be computed efficiently by precomputing chamfer distances [BTBW77] for the training images in the dataset and reducing the resolution of the query ellipses.

The proposed ellipse and label assignment proved to be robust and efficient, allowing us to fit ellipses to the user strokes and to infer their labels as the user adds new strokes to further define the pose of the object.

When the user is satisfied with the gross arrangement of masses, she can switch to contour mode. In this mode, successive pen strokes specify the boundary of the final synthesized object. Similarly to drawing the ellipses, multiple partial strokes are supported and previous strokes may be erased.

4.5. Appearance Constraints and Synthesis

Using the user's fully or partially specified pose, we can retrieve multiple training images that have similar poses. We do this by finding modes on the manifold that match the ellipses and contours with a low cost and low variance and taking the nearest neighboring training images, see Figure 7. Within this set of neighboring images, we can identify the range of possible appearances available in our dataset, for this specific pose, to use for image synthesis. We present color palettes computed from exemplars of the possible appearances to the user and allow them to select the most appropriate (see right side of Figure 1).

The data-driven nature of our algorithm means that we rely on the appearance variation in the training images to produce the appropriate variation during image synthesis.

Features The specified masses, contour strokes and appearance constraints, together with the inferred masses and contours, can be converted into features that guide the synthesis process. In addition to CIE *Lab* color channels, we have a feature channel per ellipse and another computed from the contour. The additional channels allow semantically meaningful synthesis.

Each ellipse feature channel is a truncated signed distance to the nearest point of the boundary of the ellipse, and the contour feature channel is a truncated signed distance to the nearest point of the contour. If the contour provided by the user is not closed, we estimate the most likely contour using the GP-LVM manifold. The feature channels are also computed for the nearest neighbor source images of the dataset and for the target image under the appearance constraints. Figure 8(a-b) provides examples of these feature channels.

Synthesis Synthesis of the target image is done using the image melding framework [DSB*12], using the feature channels as guiding layers, as in Image Analogies [HJO*01]. For efficiency, we use, but are not limited to, the two nearest neighbor images that are closest to the user's specifications on the GP-LVM manifold. To ensure high contrast along the contour, we double the weight of the contour feature with respect to the other features.

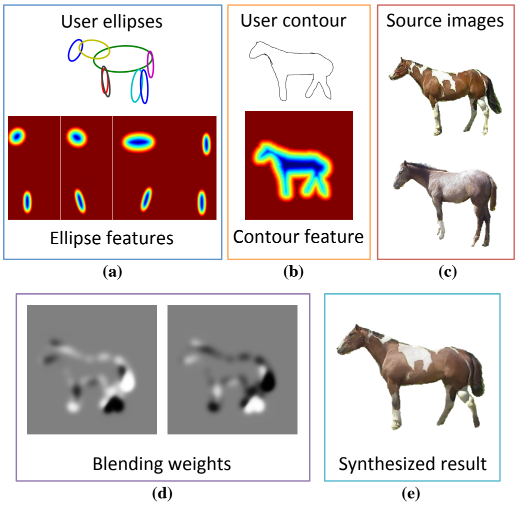


Figure 8: *Example synthesis result. (a) The ellipse configuration is used to produce a set of features (one channel per ellipse) that are combined with (b) the feature channel from the contour and (c) the CIE Lab channels of the nearest neighbor source images as an input to synthesis. (d) The blending weights for each image are computed from blurred distances between the source image feature channels and the target feature channels. (e) The synthesized result.*

At the coarsest scale of the image pyramid, we initialize the target image by computing the nearest neighbor patch correspondences using only the feature channels. Subsequent iterations use both feature channels and color channels.

At each successive scale, we compute a correspondence map from each of the source images to the target image. The target image is reconstructed using the patches of the source images according to this map, using the reconstruction costs for each source to compute blend weights; see Figure 8(d).

Figure 8(e) shows an example synthesis result from two source images. Given the large number of feature channels, the synthesis step takes around 4 minutes to perform at all scales. Thus, during the user interaction, we don't perform the synthesis up to full resolution. Instead, we produce an approximate synthesis at low resolution, and upsample the resulting nearest neighbor field to full resolution. This enables our system to efficiently synthesize a full resolution "preview" image using high resolution patches at interactive rates (about 3 seconds for an image). Although this produces some artifacts due to upsampling, the resulting preview is a reasonable proxy for the appearance of the final synthesis.

5. Synthesis Results

We have compiled 4 datasets: horses, pigeons, elephants and cats. The horse dataset was compiled using 295 images and

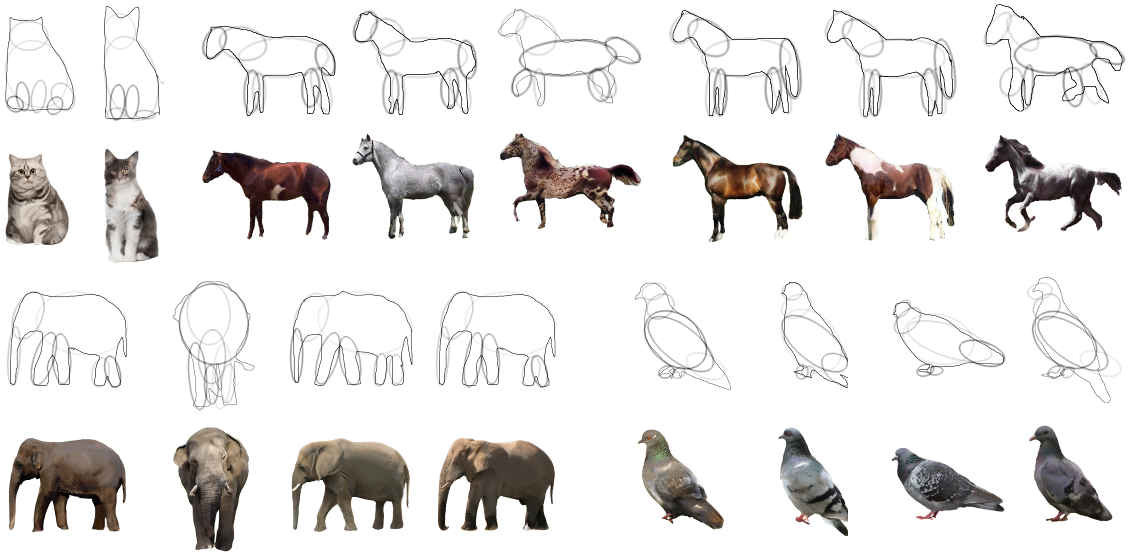


Figure 9: *Synthesis results for the horses, elephants, cats and pigeons datasets, with sketched masses and contours. Each image is a combination of training images, not simply the most similar image from a database. Note the quality of the results and their agreement with the specified masses and contours, in spite of the relatively small database sizes. At top right, the user ignored the shadow suggestions and drew a contour for the horse that is inconsistent with the masses for the front legs: This results in a synthesis failure due to incompatible constraints. See the supplemental material for details of these results.*

segmentations of the Weizmann horse dataset [BSU04]. We also collected images of pigeons (270 images) and elephants (275 images) from the internet and manually segmented the images. We have acquired a single photocollage of 390 cats on white background captured by professional photographer. All the datasets were hand-labelled into corresponding parts. Since the labeling doesn't have to be perfect, each image can be labelled in about 4 minutes. Each of the images in the dataset was rescaled, cropped and segmented from the background. The scaling was chosen such that the area of the torso matches in each image.

Figure 9 shows some results created with our system, along with the user-drawn masses and contours that produced them. Each sketch took only about 2 minutes to draw (see the supplemental video). The synthesized images appear realistic and follow the user's constraints closely. Notice that the system allows results to be produced over a wide range of poses. Please see the supplemental material for an expanded version of Figure 9 containing details of the nearest neighbor images used for the image synthesis.

Our system is not computationally expensive. The preprocess of fitting ellipses and fitting contours to the horses in the database takes 20s and 30s, respectively. Training the GP-LVM takes about 2 minutes on the same dataset. Querying the closest ellipses at run-time is interactive at 0.4s. Querying partial silhouettes takes about 0.5s. Synthesizing a preview result takes 3-4 seconds. The final, high-quality synthesis is more expensive, and usually takes around 3 minutes to compute depending on the resolution of the images.

6. User Studies

6.1. First User Study

Our system was designed to help users generate images. To assist the user in accomplishing this task, the system displays previews and shadows as a guide for the user's input. The shadow is generated both when working with ellipses and contours. Therefore, to evaluate the usefulness of our system we conduct a user study in which participants are asked to generate an image as close as possible to a target image. We evaluate the usefulness of each of the feedback visualizations: Participants perform three assignments by using three variations of our system that each employ a different set of visual feedback. For a subjective assessment of the generated image, we also assign a Manual Search assignment that asks the participant to select an image from the dataset that is closest both in terms of pose and color to the target image. Our system was designed to help users generate images.To assist the user in accomplishing this task, the system displays previews and shadows that work as a guide for the user's input. The shadow is generated both when working with ellipses and contours. Therefore, to evaluate the usefulness of our system we conduct a user study in which participants are asked to generate an image as close as possible to a target image. We evaluate the usefulness of each of the feedback visualizations: the participants perform three assignments by using three variations of our system that each employ a different set of visual feedback. For a subjective assessment of the generated image, we also assign a Manual Search assignment that asks the participant to select an image from the dataset that is

Visual Feedback	System 1	System 2	System 3
Draw Ellipses	x		
Silhouette Shadow	x	x	x
Fast NN Preview	x	x	
Coarse Preview	x	x	

Table 1: *Visual Feedbacks corresponding to the systems.*

closest both in terms of pose and color to the target image. We provide the target image to evaluate the subjective quality of the synthesized result. After completing the assignments, the participants are surveyed with the standard System Usability Scale questionnaire [Bro96] to provide a subjective assessment of general usability, as well as system-specific questions to evaluate each of the visual feedbacks' usefulness, and results of interaction with the system. We chose the horses dataset for the first user study.

6.1.1. Assignments

The Manual Search assignment requires the participant to browse through 295 images of horses to find the closest match to a target horse both in terms of pose and appearance. Each of the 295 images was rescaled, cropped and segmented from the background.

In Assignment 1, the participant uses our complete system to generate an image that is as close as possible to the target image (same target image as in the Manual Search assignment). The "System 1" used in Assignment 1 provides all visual feedbacks. Assignment 2 uses "System 2," a limited version of "System 1;" the user is not allowed to use ellipse interactions and starts with the "Draw Contour" tool. Finally, Assignment 3 uses "System 3," a further restricted system. The user is not allowed to use ellipse interactions and starts with the "Draw Contour" tool. "System 3" does not generate any kind of preview. "System 3" is similar to the Shadow Draw [LZC11] system, but with the addition of interpolated contours and an image synthesis post process. Our hypothesis is that additional visual feedback aids in the interactive image synthesis task. Table 1 shows the supported visual feedbacks for the Assignments.

6.1.2. Data Collection and Participant Selection

18 participants from the student population of our department performed the user study. Each participant was randomly assigned 3 different target images for the four assignments (Manual Search and Assignment 1 share the target image) from a set of 6 images. We did not filter the study population for handedness. Only 2 of the participants were familiar with the concept of "masses". To minimize the influence of learning effects, the assignments are conducted consecutively starting with "System 1" featuring the full set of visual feedbacks. The goal of the study was not mentioned to the participants. All participants were familiar with image editing in general and were given training using a Wacom tablet.

	Number of Votes				
System related questions	SD	D	NN	A	SA
Ellipse position feedback was useful?	0	5	1	11	1
Silhouette feedback was useful?	0	1	1	7	9
Fast "NN" preview was useful?	0	0	3	9	6
Coarse preview of the generated image with the color choices was useful?	0	0	2	9	7

Table 2: *First user study: number of votes the visual feedback related questions. The Likert response scale answers are "Strongly Disagree" (SD), "Disagree" (D), "Neither Agree Nor Disagree" (NN), "Agree" (A), "Strongly Agree" (SA).*

6.1.3. Procedure

First, we allow the user to familiarize themselves with the Wacom tablet. We allow using the mouse for the experiment, but all of the participants preferred the Wacom tablet. Before starting the tasks the participants were asked to answer two questions regarding their artistic skills and artistic training. Then, the participants were asked to perform the Manual Search assignment using a randomly assigned target image. Next, the participants were shown a video tutorial (see supplementary materials) describing the system. To clarify the part subdivision and the relationship between ellipses and parts, an example image was given to the participants. All assignments were performed without time limit. Synthesizing the final result was done on a separate machine in the background. After finishing the assignments, the participants answered the SUS questionnaire. Before conducting the system related questionnaire we recapitulate the differences between systems and show the final rendered results. Finally, we ask the participants to rate the usefulness of each of the visual feedbacks on a Likert scale (see Table 2). We provide the questionnaire in the supplementary materials.

6.1.4. Expectations

We expect the system to score above average on the SUS scale (corresponds to a SUS score above a 68). [LS09] The system was designed to assist the user through visual feedbacks, hence we expect that the participants would evaluate all visual feedbacks as "useful". We assume that "System 1" with the full set of visual feedbacks including the ellipse interactions would be evaluated as the easiest and the most efficient in accomplishing the assignment, given that it provides the most visual feedback.

6.1.5. Results

The average score of the system on the SUS scale was 68.75, which corresponds to "above average" [LS09]. Unexpectedly, both "System 1" and "System 2" received same amount of votes (9 votes each) as the easiest and the most efficient. One explanation of this result may be in the inherent preference of users to sketch without the use of masses, as we do not *teach* the users to draw using masses, and users inexperienced at drawing may not be familiar with the concept. We only show examples in the tutorial video and before Assignment 1. Moreover, the first and only trial of drawing with the masses

	Number of Votes		
Image comparison questions	Generated Image	Manual Search	System's Choice
Which one is the closest to the target image in terms of pose?	5	10	3
Which one is the closest to the target image in terms of color?	5	12	1
Which one resembles the target image the most?	7	11	0

Table 3: *First user study: number of votes for the subjective assessment of the synthesized image of the "Assignment 1".*

in Assignment 1 may be not enough to fully grasp the concept. Some of the participants said that they believe they would've performed better in Assignment 1 if they were to reuse "System 1" after completing the survey. We hypothesize that the data in Table 2 supports this point as the utility of the ellipse position feedback appears to be bi-modal with two thirds of the study participants finding the ellipse position feedback useful.

We also asked the users to compare the generated image of "Assignment 1" with their own choice from the dataset and the nearest neighbour found by the system based on the user's sketch (this ignores the appearance choice). The results can be viewed in Table 3. The generated image quality depends on the complexity and uniqueness of the target pose, the quality of the sketch and specifications, the size of the dataset etc. Nonetheless, about third of the participants found the generated image more closely resembled the target image than the manually selected image.

6.2. Second User Study

In our second user study, participants were asked to create novel images of elephants, using our elephants database containing 275 images. Whereas the horses dataset has only profile views, the elephants dataset has more diverse pose variations in 3D, including both frontal and profile views.

In the first study we provided participants with a target image in order to allow post hoc comparative evaluation of the results, but in practice, real users of our system would not have such a target image. Thus, in second study we do *not* provide participants with a target image, instead asking users to draw a sketch of an elephant pictured solely in their mind's eye. As in the first user study, the users interact with System 1 and System 2; and afterwards are surveyed with the questionaire. In this user study, we omitted the image comparison questions which are not relevant without a target image, instead focusing on the usefulness of different components of the system. Furthermore, we omit the SUS questionnaire as the usability of the system has already been evaluated in the first user study.

Table 4 shows the votes on the feedback-related questions. All participants preferred System 1, and all participants found ellipses useful for specifying pose.

	Number of Votes				
System related questions	SD	D	NN	A	SA
Ellipse position feedback was useful?	0	0	1	3	2
Silhouette feedback was useful?	0	1	0	3	2
Fast "NN" preview was useful?	1	0	2	2	1
Coarse preview of the generated image with the color choices was useful?	0	0	0	3	3
When specifying pose, ellipses were useful?	0	0	0	4	2

Table 4: *Second user study: number of votes the visual feedback related questions. The Likert response scale answers are "Strongly Disagree" (SD), "Disagree" (D), "Neither Agree Nor Disagree" (NN), "Agree" (A), "Strongly Agree" (SA).*

7. Conclusion

We have presented an interactive system for synthesizing realistic objects based on user input, given a database of training images. Our system supports a traditional illustrator's workflow, whereby the user first sketches the important masses and then refines them using contours. The advantages of this approach are the same as in traditional illustration: The gross pose of the figure can be specified loosely and iteratively, without requiring precise or complete contours. Interactive feedback is provided by indicating likely mass locations and contours to the user, as well as quickly synthesizing a preview of the object. This feedback aids novice users in understanding the pose space as they construct their sketch, and visually indicates likely outcomes for the synthesis phase. Although in this paper we demonstrate the drawing of only a few classes of objects, our general approach can be extended to other classes of objects. Positive feedback from the users indicates a promising pathway to advance interactive tools for creative illustration workflow.

7.1. Limitations and Future Work

Our present system is not without its limitations. Firstly, it is tuned for sketching animal figures. We rely on the fact that the structure of animal figures is fixed (the head is attached to the neck, and so forth), which is not true for general objects or scenes. We imagine future work might address these limitations by combining manifold methods like ours with contextual models like PatchNet [HZW*13].

We also assume that ellipses are a good representation of the shape of the body parts. Although our system gracefully handles cases where this assumption does not hold (see the horses' legs), it would be straightforward to support additional primitives to better represent a wider range of figures.

In order to synthesize realistic figures, we require the user to provide constraints (masses and contours) that are reasonably similar to some poses in the training images. If the user veers too far from the database poses, synthesis results may be unsatisfactory. An example can be seen in the top right of Figure 9, in which the pose of the foreleg masses is inconsistent with the specified contour. However, if the user allows themselves to be guided by the visual feedback, the final sketch should reside in a valid location on the manifold with sufficient training images for synthesis. In this work we err on the side of giving the user more creative control at the cost of potentially less plausible results, but it is straightforward to automatically override a user's constraints towards higher-probability locations in the manifold. This tradeoff between flexibility and plausibility warrants further exploration.

In our current system, we have not added local texture constraints, as the limited database sizes makes them hard to use. In the future, we would like to explore integrating the color and texture constraints more directly into the continuous exploration mode of the pose specification.

We currently require annotated images as input, which is labor-intensive and limits us to a sparse collection of images (on the order of a few hundred). Databases of images with semantic part labelings [CML*14, BM09, VMT*14] are becoming more widely available for addressing computer vision tasks, but in the long run, we hope to take advantage of current and future advances in computer vision to automate this part of our system [GZ12, YYB*14].

The capabilities of image synthesis algorithms are presently outstripping the interactions used to control them. We hope our system inspires further research in interactive control of structured image synthesis.

Acknowledgements The authors would like to thank Eli Shechtman for valuable discussions and Leah Anton for the sketches of horses in Figure 2. This work was funded by EPSRC grant EP/I031170/1.

References

[Bla94] BLAIR P.: *Cartoon animation (the collector's series)*. Collector's Series. Walter Foster Publishing, 1994. 3, 5

[BM09] BOURDEV L., MALIK J.: Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 1365–1372. 12

[Bro96] BROOKE J.: SUS - a quick and dirty usability scale. *Usability evaluation in industry 189* (1996), 194. 10

[BSU04] BORENSTEIN E., SHARON E., ULLMAN S.: Combining top-down and bottom-up segmentation. In *In Proceedings IEEE workshop on Perceptual Organization in Computer Vision, CVPR* (2004), vol. 4, p. 46. 9

[BTBW77] BARROW H., TENENBAUM J., BOLLES R., WOLF H.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2* (1977), pp. 659–663. 7, 8

[CCT*09] CHEN T., CHEN M.-M., TAN P., SHAMIR A., HU S.-M.: Sketch2photo: Internet image montage. *ACM Trans. on Graph. (Proc. SIGGRAPH Asia) 28*, 5 (2009). 2

[CML*14] CHEN X., MOTTAGHI R., LIU X., FIDLER S., URTASUN R., YUILLE A. L.: Detect what you can: Detecting and representing objects using holistic models and body parts. *CoRR abs/1406.2031* (2014). 12

[CTM*13] CHEN T., TAN P., MA L.-Q., CHENG M.-M., SHAMIR A., HU S.-M.: Poseshop: Human image database construction and personalized content synthesis. *IEEE Transactions on Visualization and Computer Graphics 19*, 5 (May 2013), 824–837. 2

[DPH10] DIXON D., PRASAD M., HAMMOND T.: iCanDraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), pp. 897–906. 2, 3

[Dra] DRAWINGNOW.COM: How to draw animals. http://www.drawingnow.com/how-to-draw-animals.html. [Online; accessed 05-Jan-2013]. 3, 5

[DSB*12] DARABI S., SHECHTMAN E., BARNES C., GOLDMAN D. B., SEN P.: Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. on Graph. (Proc. SIGGRAPH) 31*, 4 (2012). 2, 3, 5, 8

[ERH*11] EITZ M., RICHTER R., HILDEBRAND K., BOUBEKEUR T., ALEXA M.: Photosketcher: interactive sketch-based image synthesis. *IEEE Computer Graphics & Applications 31*, 6 (2011), 56–66. 2

[ES07] EISSEN K., STEUR R.: *Sketching: Drawing techniques for product designers*. BIS Publishers, 2007. 3

[FGF11] FERNQUIST J., GROSSMAN T., FITZMAURICE G.: Sketch-sketch revolution: An engaging tutorial system for guided sketching and application learning. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), ACM, pp. 373–382. 3

[FPF99] FITZGIBBON A., PILU M., FISHER R.: Direct least square fitting of ellipses. *IEEE Trans. on Patt. Anal. and Mach. Intell. 21*, 5 (1999), 476–480. 5

[Ful11] FULLER G.: *Start sketching and drawing now*. North Light Books, 2011. 3

[GCZ*12] GOLDBERG C., CHEN T., ZHANG F.-L., SHAMIR A., HU S.-M.: Data-driven object manipulation in images. *Computer Graphics Forum 31*, 2pt1 (2012), 265–274. 2

[GZ12] GOULD S., ZHANG Y.: PatchMatchGraph: building a graph of dense patch correspondences for label transfer. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V* (2012), pp. 439–452. 12

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), pp. 327–340. 3, 8

[HZW*13] HU S.-M., ZHANG F.-L., WANG M., MARTIN R. R., WANG J.: Patchnet: a patch-based image representation for interactive library-driven image editing. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 196. 3, 11

[IBT13] IARUSSI E., BOUSSEAU A., TSANDILAS T.: The drawing assistant: Automated drawing guidance and feedback from photographs. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (2013), pp. 183–192. 3

[JBS*06] JOHNSON M., BROSTOW G., SHOTTON J., ARANDJELOVIC O., KWATRA V., CIPOLLA R.: Semantic photo synthesis. *Comp. Graph. Forum 25*, 3 (2006), 407–413. 2

[KG82] KUHL F., GAUSSIANDINA C.: Elliptic fourier features of a closed contour. *Comp. Graph. & Img. Proc. 18*, 3 (1982). 6

[Law05] LAWRENCE N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research 6* (2005), 1783–1816. 6

[LHE*07] LALONDE J.-F., HOIEM D., EFROS A. A., ROTHER C., WINN J., CRIMINISI A.: Photo clip art. *ACM Trans. on Graph. (Proc. SIGGRAPH) 26*, 3 (2007). 2

[LS09] LEWIS J. R., SAURO J.: The factor structure of the system usability scale. In *Human Centered Design*. Springer, 2009, pp. 94–103. 10

[LZC11] LEE Y. J., ZITNICK C. L., COHEN M. F.: Shadowdraw: real-time user guidance for freehand drawing. *ACM Trans. Graph. 30*, 4 (2011), 27:1–27:10. 3, 4, 10

[McM54] MCMASTER W. H.: Polarization and the stokes parameters. *Amer. Journal of Phys. 22*, 6 (1954), 351–362. 6

[MPK09] MOHAMMED U., PRINCE S. J. D., KAUTZ J.: Visiolization: generating novel facial images. *ACM Trans. Graph. (Proc. of SIGGRAPH) 28*, 3 (2009), 57:1–57:8. 3

[NFC07] NAVARATNAM R., FITZGIBBON A., CIPOLLA R.: The joint manifold model for semi-supervised multi-valued regression. In *Computer Vision, IEEE 11th International Conference on* (2007). 6, 7

[PR11] PRISACARIU V., REID I.: Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (2011). 6

[RHDG10] RISSER E., HAN C., DAHYOT R., GRINSPUN E.: Synthesizing structured image hybrids. *ACM Trans. on Graph. (Proc. of SIGGRAPH) 29*, 4 (2010), 85:1–85:6. 3

[VMT*14] VEDALDI A., MAHENDRAN S., TSOGKAS S., MAJI S., GIRSHICK R. B., KANNALA J., RAHTU E., KOKKINOS I., BLASCHKO M. B., WEISS D., ET AL.: Understanding objects in detail with fine-grained attributes. In *IEEE Conference on Computer Vision and Pattern Recognition* (2014). 12

[XCF*13] XU K., CHEN K., FU H., SUN W.-L., HU S.-M.: Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics 32*, 4 (2013), 123:1–123:12. 2

[YYB*14] YU W., YANG K., BAI Y., YAO H., RUI Y.: Dnn flow: Dnn feature pyramid based image matching. In *Proceedings of the British Machine Vision Conference* (2014). 12