EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2006)
Thomas Stahovich and Mario Costa Sousa (Editors)

Constellation Models for Sketch Recognition

D. Sharon and M. van de Panne

University of British Columbia†

Abstract

Sketch-based modeling shares many of the difficulties of the branch of computer vision that deals with single image interpretation. Most obviously, they must both identify the parts observed in a given 2D drawing or image. We draw on constellation models first proposed in the computer vision literature to develop probabilistic models for object sketches, based on multiple example drawings. These models are then applied to estimate the most-likely labels for a new sketch. A multi-pass branch-and-bound algorithm allows well-formed sketches to be quickly labelled, while still supporting the recognition of more ambiguous sketches. Results are presented for five classes of objects.

1. Introduction

A large-class of sketch-based modeling systems, specifically those involving drawings of objects, diagrams, or maps, must solve a recognition problem. What did the user draw and what does each stroke correspond to? In many cases, this is solved with the help of domain knowledge, such as knowing that a sailboat has a mast and a hull. This recognition problem has a strong parallel with the goals of single-image interpretation in computer vision, an area which has seen significant progress over the past few years.

We apply a constellation or 'pictorial structure' model to the recognition of strokes in sketches of particular classes of objects. The model is designed to capture the structure of a particular class of object and is based on local features, such as the shape or size of a stroke, and pairwise features, such as distances to other known parts. We learn a probabilistic model from example sketches with known stroke labelings. The recognition algorithm determines a maximum-likelihood labeling for an unlabelled sketch by searching through the space of possible label assignments using a multi-pass branch and bound algorithm. Our technique supports flexible object structure by allowing for optional parts. By applying a recognition threshold, extraneous strokes can also be readily identified.

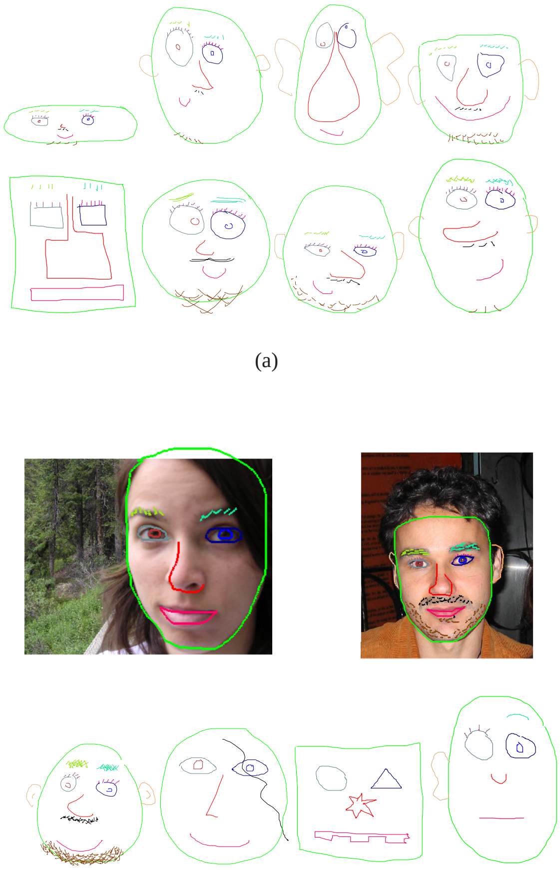
Figure 1 shows an example result for the recognition of parts in face sketches. A subset of the training examples are shown, along with a set of successfully labeled free-form sketches and trace-over sketches. A specific contribution of our method is to cope with objects that exhibit considerable variability in the way they are drawn and that allow a variable number of part instantiations.

The output of our algorithm is a set of labels assigned to the strokes. This can then be utilized by a variety of applications. Labelled strokes can be used to construct parameterized 3D models as in [YSvdP05]. Furthermore, they can help to instance models in a 2D or 3D scene, or serve as a partial interpretation of a larger sketched diagram. Sketches can also be used to retrieve images or 3D models from a database and can, in general, provide an intuitive alternative interface to models with complex internal parameterizations such as faces [fac].

Our system makes two particularly strong assumptions. First, it assumes that similar parts are drawn with similar strokes. For example, a flowerpot that is drawn with four separate strokes instead of one stroke is not easily modelled as part of the same object class. Second, object parts which are deemed mandatory in a sketch must have exactly one instance in the sketch. Optional parts may have multiple instances in a given sketch.

The remainder of the paper is organized as follows. Section 2 gives an overview of related work. Section 3 describes the details of the probabilistic constellation model. Section 4 then describes our algorithms for finding maximum-likelihood interpretations of images using the model. Results are presented and discussed in Section 5, including various

† email: dsharon,van@cs.ubc.ca



(a)

(b)

Figure 1: *(a) Face sketch training examples. The mandatory labels are* head, left-eye, right-eye, mouth, nose*; the optional labels are* left-pupil, right-pupil, left-ear, right-ear, left-eyebrow, right-eyebrow, left-eyelash, right-eyelash, moustache, beard. (b) Face sketches recognized using our system..

modes of use and examples of failure cases. Lastly, Section 6 provides conclusions and future work.

2. Related Work

In this paper we address the problem of understanding completed sketches with a known stroke structure and an unknown stroke ordering. Stroke information is assumed to be collected at the time of drawing creation or it can be extracted using image analysis of a raster drawing using morphology methods (erosion/dilation) and smooth continuation methods.

Recognizing single strokes in isolation is perhaps the simplest version of sketch understanding and can be used to support interfaces that use pen gestures as commands [Rub91]. Recognizing multi-stroke visual structure is significantly more complex, given that the interpretation of strokes is dependant on its local context. Many algorithms use some type of 'parse tree' to search through the space of possible stroke labelings in order to find the most consistent interpretation of a given set of strokes. For applications that involve diagram interpretation, the search is often anchored by first finding well-defined symbols, such as drawn characters or electrical component symbols [KS04]. The search is then further constrained by exploiting the known structure of the given application domain or object classes.

Matching can be treated as a graph isomorphism problem [MF02], where it is applied to the recognition of human stick figures using a known model of connectivity. The work of [YSvdP05] applies a flexible form of hierarchical graph matching. For example, it first looks for the best subgraph representing a cup body before then proceeding to look for optional parts such as cup handles. Curve shape feature vectors are used to quantify the best match and stochastic search is used to explore the space of possible matches. Both of these graph-based models rely heavily on connectivity between parts. They are thus weak at recognizing drawings with disjoint parts, such as a nose or an airplane window.

A probabilistic approach to sketch stroke interpretation is proposed in [AD04]. This uses domain-specific libraries of 'Bayesian network fragments' that describe shapes and domain patterns. Several mechanisms to control the size of the space of hypotheses are presented, and the technique is applied to the domain of electrical circuit diagram recognition. [QSM05] proposes the use of conditional random fields for labeling box-and-line diagrams for particularly difficult ambiguous examples where constraints must propagate in order to find the most-likely interpretation. Perceptually-based shape descriptions are used to help infer the the recognition of image structure in [SMF*02]. Our work looks at recognition problems that do not require connectivity between parts and considers object sketches that can exhibit considerable variability.

Image-based techniques can also be used to help identify sketches or parts of sketches. Shape contexts [BM02] can be used to match sketch images to a fixed set of prototype template images. Image-based classifiers are applied in [SV04] in order to determine likely interpretations for subsets of strokes. An A* search procedure is used to search among the space of possible subset of strokes in order to find a maximum-likelihood interpretation for the image. This is applied to a graphic symbol set of 13 symbols.

Constellation models, also known as pictorial structure models, are composed of a set of local parts, each of which has an appearance model, and a geometry model that defines preferred relative locations or distances of the parts [FE73]. They are well suited to applications such as face recognition, where features such as the nose, eyes, and mouth have particular local features and also have relatively well-defined distances to each other. The model is further developed in [FH05], where it is applied to identify both faces and body configurations from images. The model continues to be extended, with an emphasis on learning pictorial structure models automatically from example images of object classes. More generally, this can be viewed as an example of *statistical relational learning*.

An agent-based approach is presented in [MA03], although this relies on a predefined grammar for the description of the components. The work of [KLP05] is similar to ours in that it uses a constellation-type model and a probabilistic framework. Our work differs in a number of respects, including application to a different domain, using different and larger individual and pairwise feature sets, supporting flexible object classes with optional parts, and a staged search strategy.

Our approach for sketch recognition is uniquely characterized by: (a) support for model definitions derived directly from a set of drawn training examples; (b) a probabilistic framework; (c) support for optional parts; (d) a constellation model with features specifically suited for sketch recognition; and (e) an efficient multi-stage search strategy. We demonstrate our approach on five classes of objects and multiple modes of use (drawing and tracing).

3. The Constellation Model

We represent an object using a constellation model, consisting of features of individual object parts, as well as features of pairs of parts. Individual features capture shape and global positions of parts, whereas pairwise features summarize relative positions of parts. An example constellation model of a face object is shown in Figure 2.

We define a four-element feature vector for individual object parts: $\mathcal{F} = [x\ y\ d\ \beta]$ where (x,y) are the location of the center of the axis-aligned bounding-box (AABB) of a stroke, as measured in image coordinates normalized to $x,y \in [0,1]$; d is the normalized-coordinate length of the AABB diagonal; and $\beta = cos(\phi)$, with ϕ being the angle of the AABB diagonal with respect to the x-axis.

Similarly, we choose a four-element feature vector for part pairs defined by $\mathcal{G}_{ab} = [\Delta x_{ab}\ \Delta y_{ab}\ D_{ab}\ D_{ba}]$, where $\Delta x = x_a - x_b$ and $\Delta y = y_a - y_b$ define the relative positions of the AABB centers of strokes a and b in normalized coordinates, D_{ab} is the minimum distance between the endpoints of stroke a and any point on stroke b, and D_{ba} is the minimum distance between the endpoints of stroke b and any point on stroke a. In general, $\mathcal{G}_{ab} \neq \mathcal{G}_{ba}$.

Full constellation models do not scale well with the number of parts, n, since they result in $O(n^2)$ pairwise features. We choose to alleviate this by characterizing each label as mandatory or optional. Individual features are computed for both mandatory and optional parts. However, pairwise features are only computed if one or both of the labels in the pair corresponds to a mandatory part. We note that it may be possible to further reduce the number of pairwise features by searching for subsets that yield good recognition performance [CFH05].

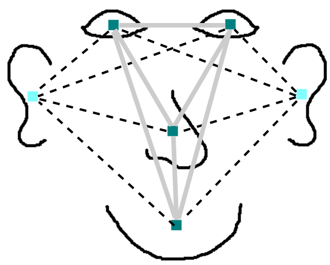


Figure 2: *Example constellation model for a sketched face, showing the pairwise interactions. In this example, the left-eye, right-eye, mouth, and nose are mandatory and thus have complete pairwise interactions. The left-ear and right-ear are optional and thus have pairwise interactions with all mandatory parts but not with each other.*

The sketch recognition process has two phases, the first which searches the space of possible mandatory label assignments, and the second which searches for optional labels for the remaining unlabelled strokes. In this way the mandatory labels provide contextual location information necessary for assigning appropriate labels to the potentially large number of optional parts. We describe the search algorithm in the following section.

3.1. Learning the Model

An object class model is represented using a probability distribution over features in object constellation models. This function is learned from a set of example labelled sketches. A straightforward choice of object class model is to use multivariate Gaussian distributions. However, in order to support recognition from a small number of training examples, we opt for a diagonal covariance matrix. Thus we independently compute the mean and covariance of each element of the feature vectors $\mathcal{F}$ and $\mathcal{G}$ for the set of labelled sketches that serve as training data. More explicitly, the probabilistic model for the fth element in the feature vector of a label ℓ is given by θ_ℓ^f and consists of the mean value for the feature element, μ_ℓ^f, as well as the standard deviation, σ_ℓ^f. Similarly, a pair feature model, $\theta_{\ell j}^f$, is given by $< \mu_{\ell j}^f, \sigma_{\ell j}^f >$.

3.2. Labeling Likelihood

The quality of a particular matching between labels and strokes is scored using a cost function. In early constellation models, the match quality is defined in terms of an energy that is a function of both the individual feature matches and the pairwise feature match. As in other recent work [CFH05], we cast the problem in a probabilistic framework and search for the most likely interpretation. The probability of a given labeling L is given by the product of the individual stroke labeling likelihoods, further multiplied by the product of all labelled stroke pair likelihoods. This can be expressed as:

$$P(L|\theta) = \prod_{i=1}^{N}\prod_{\ell=1}^{M} P(\mathcal{F}_i|\theta_\ell)^{\delta_{i\ell}} \prod_{j=1}^{m}\prod_{k=1}^{N} P(\mathcal{G}_{ik}|\theta_{\ell j})^{\delta_{kj}} \qquad (1)$$

In the above expression, the interior of the first term, $P(\mathcal{F}_i|\theta_\ell)^{\delta_{i\ell}}$, represents the probability of stroke i having label ℓ. This is computed for all strokes, as given by the outside product. The inside product is a notational convenience for expressing the stroke-label assignment.

The interior of the second term, $P(\mathcal{G}_{ik}|\theta_{\ell j})^{\delta_{kj}}$, represents the probability of stroke i in relation to all the mandatory parts, as measured by the pairwise feature vectors. Thus if a stroke is labelled as right ear but it is located below the mouth, then it is this term that will give that labeling a low likelihood. Pairwise relations are computed with respect to all mandatory parts, as given by the outside product. The inside product is a notational convenience for expressing the stroke-label assignment for the mandatory strokes.

The assignment of strokes to labels is modelled as a label-assignment matrix δ, with $\delta_{i\ell} = 1$ if stroke i is assigned label ℓ, and $\delta_{i\ell} = 0$ otherwise. The exponentiation using the δ values is a notational convenience for compactly representing stroke-label assignments. All terms having an exponent of $\delta = 0$ evaluate to 1 and thus effectively drop out of the likelihood computation. The label-assignment matrix has imposed upon it the appropriate restriction that each stroke can be assigned only one label, and that mandatory labels should map to a unique stroke. N is the number of strokes, M is the number of labels, and m is the number of mandatory labels. $P(\mathcal{F}_i|\theta_\ell)$ models the likelihood of stroke i having label ℓ. Similarly, $P(\mathcal{G}_{ik}|\theta_{\ell j})$ models the likelihood of the stroke pair (i,k) having the labeling (ℓ, j). The above omits the normalizing constant $P(\theta)$, which does not affect the ML solution. We assume a uniform prior on the likelihood of parts appearing in a sketch.

4. Maximum Likelihood Search

A maximum likelihood (ML) search procedure finds the most plausible labelling for all strokes that appear in the image. For a simple application of a constellation model having n strokes and m independent object part labels, there are m^n possible assignments that could in principle be explored, and each assignment configuration requires evaluating $O(n^2)$ pairwise interactions. Further complications arise because some strokes may not have plausible labels, and some object parts (i.e., labels) may not be found in a given sketch, or may have multiple instances. In order to allow for these complications, and to alleviate the computational cost associated with the exponential number of matches, the search over possible label assignments has two phases.

The first search phase involves labelling strokes that correspond only to the mandatory object parts and then commiting to those labels. This is followed by a linear search through the optional labels for the recognition of the remaining unlabelled strokes. Both search phases use the same objective function, namely the likelihood as described in the previous section.

The search over possible label assignments is carried out using a branch-and-bound search tree. Each node in the search tree represents a partial labeling of the sketch. A node at depth i in the tree has found corresponding strokes for labels 1 through i. Each node in the tree has a current assigned likelihood which is determined from the product of individual stroke-label likelihoods for the i assigned labels, as well as all the pairwise interaction likelihoods among all labelled parts.

To advance the search, a node is extended by evaluating all possible assignments of mandatory label $i+1$ to unlabelled strokes. During the search, the algorithm tracks the cost of the best (most likely) known complete assignment of mandatory labels. The cost is used to bound branches of the search. Each completed search branch can potentially result in a better bound to restrict the remaining search.

Branches of the search tree can only be bounded once a complete assignment of mandatory labels is found. If the number of strokes or mandatory labels is high, finding complete assignments is prohibitively slow. We employ two approaches to further constrain the search: multipass thresholding and hard constraints.

With multipass thresholding, we bound branches of the search before encountering a full labelling. If a node's likelihood, as computed by its current partial set of label assignments, is lower than a specified threshold α, that search branch is terminated. We use multiple passes, beginning with an optimistic threshold. That is, at first, we assume all feature likelihoods in a match will be very high. This can result in an overly restrictive search that may lead to no complete labelings being found. However, this is quick to compute in comparison to a full search, or a search with a more pessimistic bound.

Upon failure to find a successful complete label assignment, each successive pass of the branch-and-bound search uses a progressively more pessimistic assumption until complete solutions are found. The first complete solutions found are then used as a good bound for a final search pass wherein the threshold can be as pessimistic as is desired. We begin with a threshold corresponding to $P(\mu + 1.3\sigma)$ for each feature element likelihood, and on each successive pass we scale this by $2/3$. Multi-pass thresholding makes the search feasible for a large number of strokes and mandatory labels and also results in fast labeling for 'good' sketches while supporting more extensive searches through the hypothesis space for assigning labels to more ambiguous sketches.

In lieu of a threshold based on the likelihood-to-date for the partial assignments, an alternative that we have found to be equally successful is to threshold based on individual part and pair likelihoods. Thus, a branch is terminated when it involves any individual likelihood that falls below a threshold β. For the examples shown in the paper, this is the type of multi-pass thresholding that we apply.

Hard constraints can be seen as a variant on the type of thresholding just described. For a particular object class, it may be the case that one feature label should always satisfy a particular relation with respect to another. For example, the nose could be required to always be located above the mouth in a face sketch. For our implementation, we infer *above, below, left,* and *right* relationships from the example sketches wherever they can be found. Thus, if the nose AABB center appears above the mouth AABB center in all the example sketches, this will be added as a hard constraint. An object class may have many such constraints between labelled parts.

5. Results and Discussion

We have tested the method on the 5 classes of objects listed in Table 1. These have 7–15 labels and have been tested on drawings having 3–200 strokes. We use on the order of 20-60 training examples for each class. Figures 1, 3, 4, 5, and 6 show training sketches and successful test sketches. The recognition time is typically 0.01-2.5s for the shown examples, with most of this time being spent on initialization. During initialization, a feature vector $\mathcal{F}$ is pre-computed for all strokes and another feature vector $\mathcal{G}$ for all stroke pairs of the input sketch. As an example, consider the bottom left face sketch in Figure 1, which contains 171 strokes. The recognition takes a total of 1.97 seconds, with 80% of the computation time spent on initialization, 18% on searching for mandatory labels, and 2% on finding labels for the optional parts. Spurious strokes can be rejected by placing a threshold on the fit of optional stroke labels.

The hard constraints discussed in Section 4 may significantly reduce recognition times. However, when they are automatically inferred from training data, the system may falsely register the existence of a hard constraint. For example, few training sketches may result in the system falsely believing that the left eye is always below a right eyelash. However, such a situation could easily occur in a cartoon-style face sketch or a somewhat asymetric sketch. This could be viewed as an indication that more training data is required.

Figure 7 shows a set of failure examples, meaning that one or more strokes are mislabeled. Recognition can go wrong in several ways: (1) inability to find suitable mandatory strokes because of the hard constraints; (2) mislabeling of a mandatory stroke, leading to havoc with the remaining strokes; (3) mislabeling of optional strokes. In practice, errors of type (1) are rare and imply a lack of training data. Errors of type (2) can occur if unusual strokes occur that affect the overall bounding box and therefore result in atypical normalized coordinates. This might occur for adding overly long or bushy hair in face sketches, or certain atypical stems in flower sketches. Errors of type (3) most commonly occur when there are few mandatory strokes, such as for the sailboats or flowers. The model does not currently give any consideration to relationships between optional parts. Lastly, other mislabelings can be attributed to impoverished probability distribution models and inadequate feature vectors.

class	mandatory labels	optional labels
faces	5	10
flowers	2	5
sailboats	3	5
airplanes	3	4
characters	7	8

Table 1: *Object classes.*

In order to evaluate the utility of the multipass thresholding technique, we test the recognition of sketches with and without thresholding. Table 2 shows the results of this experiment. In all cases, the multipass thresholding results in significantly lower computation times. Most notable is a 103-stroke face sketch which took only 1.242 seconds to recognize with thresholding, yet without thresholding, failed to find a labeling within 9 hours.

class	num strokes	with multipass (s)	without multipass (s)
face	103	1.242	> 9 hours
flower	54	0.46	0.98
sailboat	8	0.02	0.03
airplane	21	0.08	0.1
character	18	0.12	126.69

Table 2: *Computation times for recognition algorithm with and without the multi-pass technique.*

Our system assumes a uniform prior for the a priori likelihood of optional parts. This decision stems in part from our expectation that a small number of training sketches will not necessarily reflect the probability of parts appearing in future sketches. Thus, the identity of parts depends solely on their shape and fit as modeled by the constellation model.

(a)

(b)

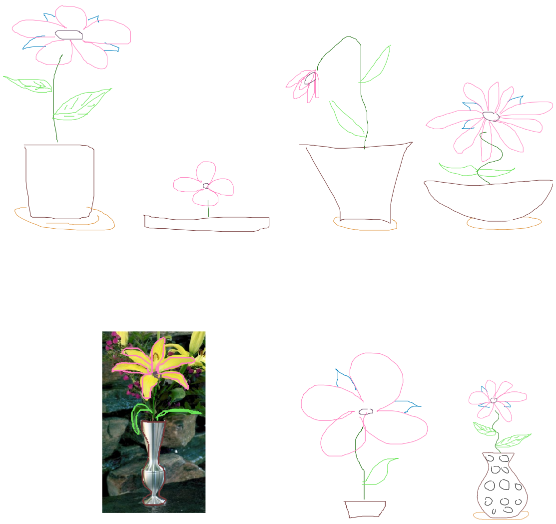


Figure 3: *(a) Flower sketch training examples. The mandatory labels are* pot, stem, stigma*; the optional labels are* sacuer, petal, leaf, and sepal*. (b) Flower sketches recognized using our system.*

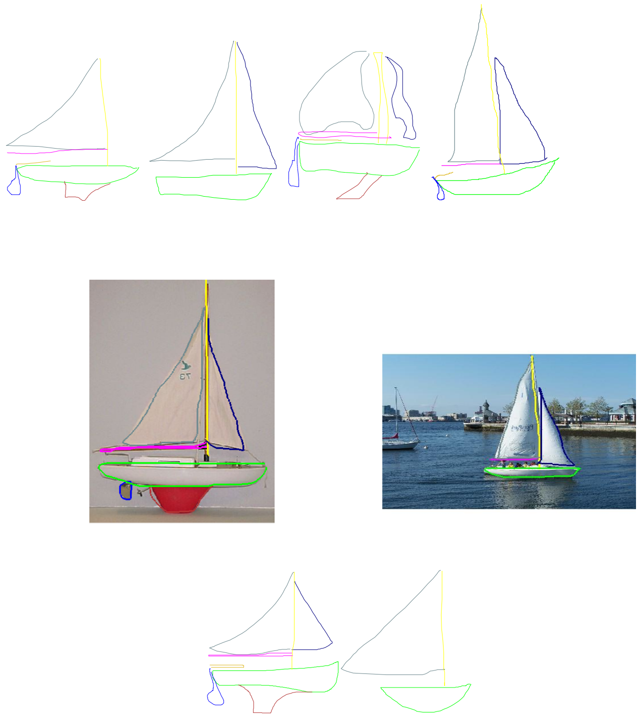


Figure 4: *(a) Sailboat sketch training examples. The mandatory labels are* hull, main-sail, mast*; the optional labels are* jib, boom, keel, rudder, tiller*.(b) Sailboat sketches recognized using our system.*

We define the recognition of a sketch to be the labeling of the individual parts of a sketch that is of a known object class. If the class of the input sketch is unknown, the maximum likelihood fit could be determined for each of a list of object classes in order to provide information about the object class. The ML log-likelihoods that come from each class are not directly comparable, however, because object classes differ in their number of mandatory parts. Mandatory parts have fully-connected pairwise likelihoods while optional parts only have pairwise likelihoods in relation with mandatory parts. An appropriate normalization can be constructed to deal with this, although we have yet to investigate this. We believe that there are likely better discriminative object-classification methods that do not rely on complete part labeling.

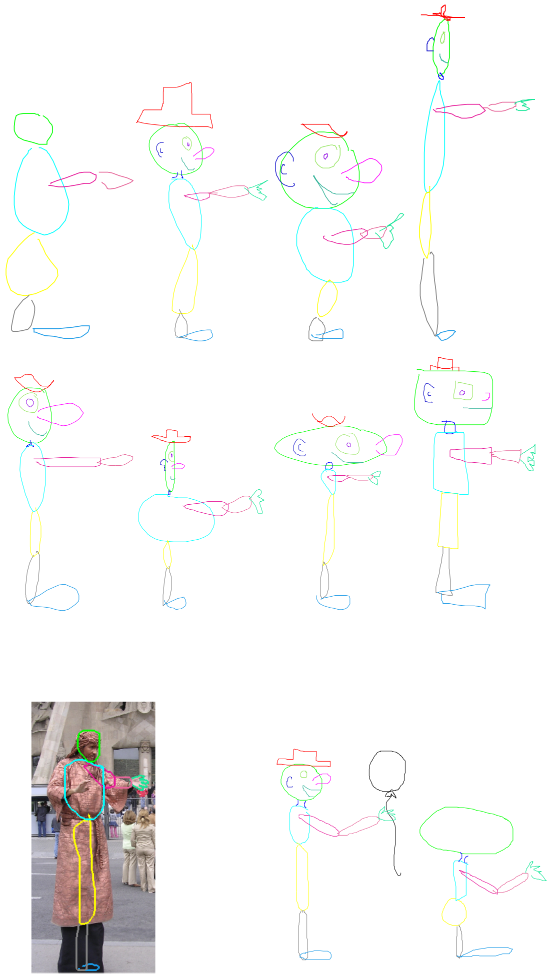
It may be possible to further improve on the mean search time for the branch-and-bound algorithm by using variants of the A* algorithm. This involves expanding non-terminal nodes in the search in an order sorted by their cost-to-date. However, much of the leverage of A* comes from the ability to generate a suitable always-optimistic cost-to-go function. Unfortunately this provides little leverage given that it is possible that the remaining unlabelled strokes could perfectly match the mean features.

6. Conclusions

We have presented a system that adapts *constellation* or *pictorial structure* models from the computer vision literature for flexible sketch recognition. Adaptations include support for optional parts, the use of an efficient multi-pass branch and bound search for exploring the space of possible interpretations, and the construction of individual and pairwise features suitable for sketch recognition.

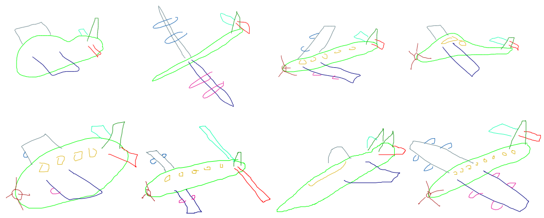
There remain a number of open directions for future work. The most significant limitation of the current system is the requirement to have one label per stroke. Thus, an eye or plane wing always needs to be drawing with a single stroke in our system. The ideas presented in [MA03] could be used to dynamically instantiate ‘part hypotheses’, which provide a path for top-down knowledge to help assemble multiple local strokes into a single primitive. Local stroke proximity information [SV04] may also be useful in determining likely groupings of strokes that represent a single part. It may also be possible to encode the most common stroke patterns used to construct a part into multiple part templates that are all associated with the same label. Lastly, a gestalt-based bottom-up grouping process may provide a significant speedup when assigning identical labels to large groups of strokes.

The current recognition process is largely top-down: the



(b)

Figure 5: *Character sketch training examples. The mandatory labels are* head, torso, thigh, shin, foot, upper-arm, lower-arm*; the optional labels are* hat, neck, hand, nose, eye, pupil, mouth, ear. *(b) Character sketches recognized using our system.*



(a)

(b)

Figure 6: *(a) Airplane sketch examples. The mandatory labels are* fuselage, left-wing, right-wing*; the optional labels are* left-stabilizer, right-stabilizer, left-engine, right-engine, propellor, window, tail-fin. *(b) Airplane sketches recognized using our system.*

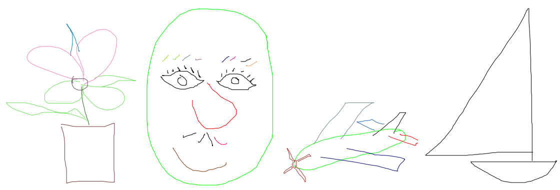


Figure 7: *Failure modes of our system. Mislabelling can be caused by lack of training sketches and inadequate features.*

diagram is searched for mandatory labels in a fixed order in order to construct the search tree. This ignores bottom-up information that could be used to reorder the search to begin with the strokes which have likely bindings to particular labels, thereby strongly constraining the search early on.

In the current system we have only experimented with a limited number of individual and pairwise features. It is likely that features other than those we have proposed will be useful in producing a more robust system. Given a large set of possible features and appropriate datasets, it should be possible to run an offline process that determines the k most informative features (individual and pairwise). How to best represent the probability distributions for a given set of features is a further open problem. Our model of independent, normally-distributed features is well suited for systems relying on only a small set of example labelled sketches. However, multi-variate Gaussian models or mixtures of Gaussians may provide better results for larger data sets, at the expense of requiring a larger number of labelled examples.

Acknowledgements

The authors would like to thank Kevin Murphy for early discussions related to this project, and the anonymous reviewers for useful suggestions for improving the paper.

References

[AD04] ALVARADO C., DAVIS R.: Sketchread: a multi-domain sketch recognition engine. In *UIST '04 ACM symposium on User interface software and technology* (2004), pp. 23–32.

[BM02] BELONGIE S., MALIK J.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 24 (April 2002), 509–522.

[CFH05] CRANDALL D., FELZENSZWALB P. F., HUTTENLOCHER D. P.: Spatial priors for part-based recognition using statistical models. In *Proceedings of CVPR* (2005), pp. 10–17.

[fac] Facegen. Singular Inversions Inc., http://www.facegen.com.

[FE73] FISCHLER M. A., ELSCHLAGER R. A.: The representation and matching of pictorial structures. *IEEE Transactions on Computers 22*, 1 (1973).

[FH05] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Pictorial structures for object recognition. *Intl. Journal of Computer Vision 61*, 1 (January 2005), 55–79.

[KLP05] KAELBLING L., LOZANO-PEREZ T.: *Learning Three-Dimensional Shape Models for Sketch Recognition*. Tech. rep., MIT CSAIL, Jan 2005.

[KS04] KARA L. B., STAHOVICH T. F.: Hierarchical parsing and recognition of hand-sketched diagrams. In *Proceedings of UIST'04* (2004).

[MA03] MACKENZIE G., ALECHINA N.: Classifying sketches of animals using an agent-based system. In *Proceedings of the 10th International Conference CAIP, Springer Lecture Notes in Computer Science 2756* (2003), pp. 521 – 529.

[MF02] MAHONEY J. V., FROMHERZ M. P. J.: Three main concerns in sketch recognition and an approach to addressing them. In *AAAI Spring Symposium on Sketch Understanding* (2002).

[QSM05] QI Y., SZUMMER M., MINKA T. P.: Diagram structure recognition by bayesian conditional random fields. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

[Rub91] RUBINE D.: Specifying gestures by example. In *SIGGRAPH '91* (1991), pp. 329–337.

[SMF*02] SAUND E., MAHONEY J., FLEET D., LARNER D., LANK E.: Perceptual organization as a foundation for intelligent sketch editing. In *AAAI Spring Symposium on Sketch Understanding* (2002).

[SV04] SHILMAN M., VIOLA P.: Spatial recognition and grouping of text and graphics. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2004).

[YSvdP05] YANG C., SHARON D., VAN DE PANNE M.: Sketch-based modeling of parameterized objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005).