Finding Objects by Grouping Primitives

David Forsyth, John Haddon, and Sergey Ioffe

Computer Science Division, U.C. Berkeley, Berkeley, CA 94720, USA
daf,ioffe,haddon@cs.berkeley.edu,
WWW home page: http://www.cs.berkeley.edu/~daf,ioffe,haddon

Abstract. Digital library applications require very general object recognition techniques. We describe an object recognition strategy that operates by grouping together image primitives in increasingly distinctive collections. Once a sufficiently large group has been found, we declare that an object is present. We demonstrate this method on applications such as finding unclothed people in general images and finding horses in general images. Finding clothed people is difficult, because the variation in colour and texture on the surface of clothing means that it is hard to find regions of clothing in the image. We show that our strategy can be used to find clothing by marking the distinctive shading patterns associated with folds in clothing, and then grouping these patterns.

1 Background

Several typical collections containing over ten million images are listed in [6]. There is an extensive literature on obtaining images from large collections using features computed from the whole image, including colour histograms, texture measures and shape measures; significant papers include [9, 13, 16, 21, 24, 25, 27, 30, 31, 36–39, 42].

However, in the most comprehensive field study of usage practices (a paper by Enser [6] surveying the use of the Hulton Deutsch collection), there is a clear user preference for searching these collections on image semantics; typical queries observed are overwhelmingly oriented toward object classes ("dinosaurs", p. 40, "chimpanzee tea party, early", p. 41) or instances ("Harry Secombe", p. 44, "Edward Heath gesticulating", p. 45). An ideal search tool would be a quite general recognition system that could be adapted quickly and easily to the types of objects sought by a user. Building such a tool requires a much more sophisticated understanding of the process of recognition than currently exists. Object recognition will not be comprehensively solved in the foreseeable future. Solutions that are good enough to be useful for some cases in applications are likely, however. Querying image collections is a particularly good application, because in many cases no other query mechanism is available — there is no prospect of searching all the photographs by hand. Furthermore, users are typically happy with low recall queries - in fact, the output of a high-recall search for "The President" of a large news collection would be unusable for most application purposes. This proposal focuses on areas that form a significant subset of these queries where useful tools can reasonably be expected.

Discussing recognition requires respecting a distinction between two important and subtly different problems: *finding*, where the image components that result from a single object are collected together; and *naming*, where the particular name of a single isolated object is determined. Finding is not well defined, because objects are not well defined — for example, would one regard the image components corresponding to an ear or an eye as separate objects that comprise a face, or do these components belong together as part of a single indissoluble object?

2 Primitives, segmentation and implicit representations

Writings on object recognition have tended to concentrate on naming problems. For some types of object or scene finding can be avoided by quite simple techniques. For example, for small numbers of geometrically exact object models search is effective [7, 14, 18, 22, 26, 28, 29, 34, 40]; and for isolated objects, finding is irrelevant.

However, in many applications finding is an important component of the problem; often, the name of an object is required only at a very limited level of detail ("person", "big cat", etc.). While naming is not an easy problem, quite good solutions appear possible with extensions of current pose-based techniques. There are several reasons finding is very difficult and poorly understood. Finding is essentially segmentation writ large, using generic cues — like coherence in colour and texture, used by current work on segmentation — initially and high-level knowledge later to obtain *regions that should be recognised together*. However, deciding which bits of the image belong together and should be recognised together requires knowledge of object properties. As a result, finding involves deploying object knowledge to direct and guide segmentation — but how is the right piece of knowledge to be used in the right place? One wishes to recognize objects at a class level independent of geometric detail, so that finding algorithms should be capable of **abstraction**. For example, most quadrupeds have roughly the same body segments in roughly the same place — good finding algorithms would exploit this fact before, say, measuring the distribution of musculature on each segment or the number of hairs on an ear. Finally, a sensible approach to finding should use representations that are robust to the effects of **pose**, and of **internal degrees of freedom**, such as joints.

If we use the word primitive more loosely, to mean a feature or assembly of features that has a constrained, stylised appearance, then a representation based around primitives at many levels has the great advantage that, at each stage of finding, a program can know what it is looking for. For example, horses can be represented (crudely!) as assemblies of hide-coloured cylinders — this results in a finding process that first looks for hide-like regions; then finds edge points, and uses geometrical constraints to assemble sets of edge points that could have come from cylinders; and finally reasons about the configuration of the cylinders. At each stage there are few alternatives to choose from, which means the search is efficient; and, while each individual test is weak, the collective of tests in sequence

can be quite powerful. The choice of primitives and the order and nature of assembly routines together form an *implicit representation* — a representation of an object as a finding process which functions as a source of top-down knowledge. We now have some insight into what should be a primitive. Primitives should have **stereotyped appearance**. The most useful form of primitive is one where it is possible to test an assembly of image features and say whether it is likely to have come from a primitive or not. For example, it is known that such tests are easy for surfaces of revolution, straight homogeneous generalised cylinders, canal surfaces, and cylinders [32, 33, 43]. As a result, it is possible to segment image regions that are likely to correspond to such surfaces *without knowing to what object they belong*[1]. A second feature of a useful primitive is that it is **significant**. For example, a cylinder is a significant property, because many objects are - at a crude level - made of cylinders. A third useful property is **robustness**; cylindrical primitives are quite easy to find even in the presence of some deformations. These properties mean that finding objects that are assemblies of primitives essentially involves finding the primitives, and then reasoning about their assembly. As we have indicated, previous work has typically concentrated on parsing activities (which assume that finding has already occurred); this proposal concentrates on finding.

2.1 Body plans - interim results on implicit representations

A natural implicit representation to use for people and many animals is a *body plan* — a sequence of grouping stages, constructed to mirror the layout of body segments. These grouping stages assemble image components that could correspond to appropriate body segments or other components (as in figure 1, which shows the plan used as an implicit representation of a horse). Having a sequence of stages means the process is efficient: the process can start with checking individual segments and move to checking multi-segment groups, so that not all groups of four (or however many for the relevant body plan) segments are presented to the final classifier. We have done extensive experiments with two separate systems that use the same structure:

- Images are masked for regions of appropriate colour and texture.
- Roughly cylindrical regions of appropriate colour and texture are identified.
- Assemblies of regions are formed and tested against a sequence of predicates.

The first example identifies pictures containing people wearing little or no clothing, to finesse the issue of variations of appearance of clothing. This program has been tested on an usually large and unusually diverse set of images; on a test collection of 565 images known to contain lightly clad people and 4289 control images with widely varying content, one tuning of the program marked 241 test

[1] While current techniques for finding generalised cylinders are fragile, because they winnow large collections of edges to find subsets with particular geometric properties and so are overwhelmed by images of textured objects, the principle remains. We indicate an attack on this difficulty below.

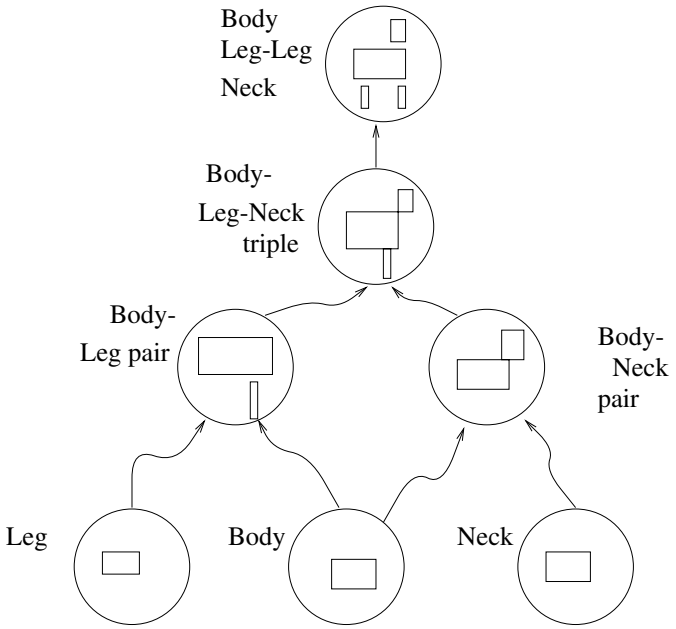


Fig. 1. *The body plan used for horses. Each circle represents a classifier, with an icon indicating the appearance of the assembly. An arrow indicates that the classifier at the arrowhead uses segments passed by the classifier at the tail. The topology was given in advance. The classifiers were then trained using image data from a total of 38 images of horses.*

images and 182 control images (the performance of various different tunings is indicated in figure 3; more detailed information appears in [12, 10]). The recall is comparable with full-text document recall [3, 4, 35] (which is surprisingly good for so abstract an object recognition query) and the rate of false positives is satisfactorily low. In this case, the representation was entirely built by hand.

The second example used a representation whose combinatorial structure — the order in which tests were applied — was built by hand, but where the tests were learned from data. This program identified pictures containing horses, and is described in greater detail in [11]. Tests used 100 images containing horses, and 1086 control images with widely varying content. The geometric process makes a significant different, as figure 2 illustrates. The performance of various different configurations is shown in figure 3. For version "F", if one estimates performance omitting images used in training and images for which the segment finding process fails, the recall is 15% — i.e. about 15% of the images containing horses are marked — and control images are marked at the rate of approximately

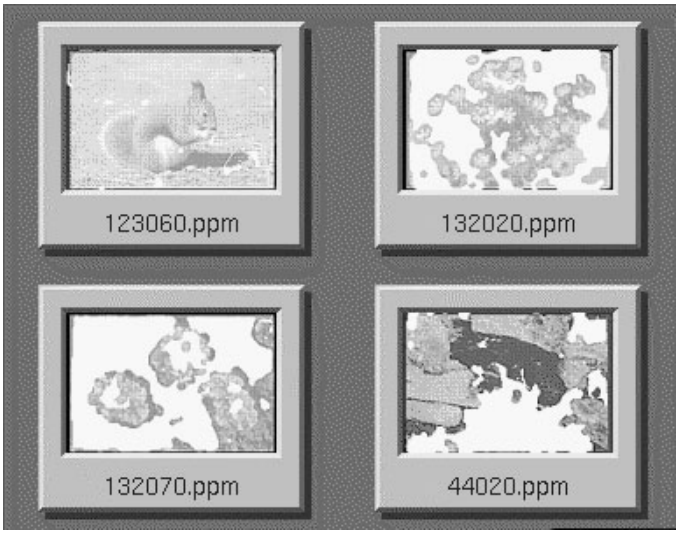


Fig. 2. *Typical images with large quantities of hide-like pixels (white pixels are not hide-like; others are hide-like) that are classified as* not *containing horses, because there is no geometric configuration present. While the test of colour and texture is helpful, the geometric test is important, too, as the results in figure 3 suggest. In particular, the fact that a horse is brown is not nearly as distinctive as the fact that it is brown, made of cylinders, and these cylinders have a particular set of possible arrangements.*

0.65%. In our test collection, this translates to 11 images of horses marked and 4 control images marked[2].
Finding using body plans has been shown to be quite effective for special cases in quite general scenes. It is relatively insensitive to changes in aspect [11]. It is quite robust to the relatively poor segmentations that our criteria offer, because it is quite effective in dealing with nuisance segments — in the horse tests, the average number of four segment groups was $2,500,000$, which is an average of forty segments per image. Nonetheless, the process described above is crude: it is too dependent on colour and texture criteria for early segmentation; the learning process is absent (humans) or extremely simple (horses); and there is one recogniser per class.

3 Learning assembly processes from data

We have been studying processes for learning to assemble primitives. The recognition processes described above have a strong component of correspondence; in particular, we are pruning a set of correspondences between image segments and body segment labels by testing for kinematic plausibility.

[2] These figures are *not* 15 and 7, because of the omission of training images and images where the segment finder failed in estimating performance.

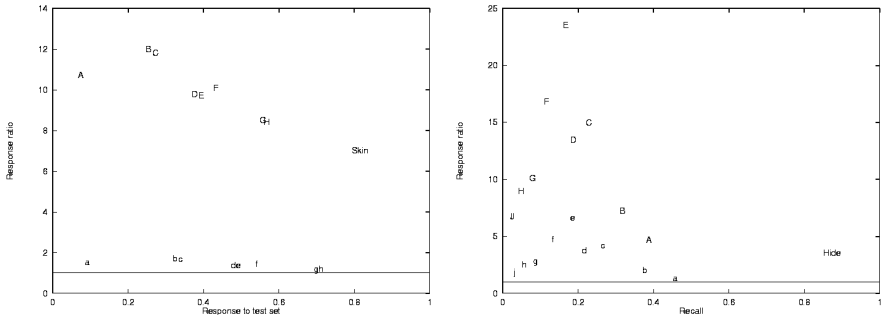


Fig. 3. *The response ratio, (percent incoming test images marked/percent incoming control images marked), plotted against the percentage of test images marked, for various configurations of the two finding programs. Data for the nude human finder appears on the top, for the horse finder on the right. Capital letters indicate the performance of the complete system of skin/hide filter and geometrical grouper, and lower case letters indicate the performance of the geometrical grouper alone. The label "skin" (resp "hide") indicates the selectivity of using skin (resp hide) alone as a criterion. For the human finder, the parameter varied is the type of group required to declare a human is present — the trend is that more complex groups display higher selectivity and lower recall. For the horse finder, the parameter being varied is the maximum number of that will be considered.*

The search for acceptable correspondences can be made efficient by using *projected classifiers*, which prune labelings using the properties of smaller sublabelings (as in [18], who use manually determined bounds and do not learn the tests). Given a classifier C which is a function of a set of features whose values depend on segments with labels in the set $L = \{l_1 \ldots l_m\}$, the projected classifier $C_{l_1 \ldots l_k}$ is a function of of all those features that depend only on the segments with labels $L' = \{l_1 \ldots l_k\}$. In particular, $C_{l_1 \ldots l_k}(L') > 0$ if there is some extension L of L' such that $C(L) > 0$. This criterion corresponds to insisting that groups should pass intermediate classifiers if, *with appropriate segments attached*, they pass a final classifier.

The converse need not be true: the feature values required to bring a projected point inside the positive volume of C may not be realized with any labeling of the current set of segments $1, \ldots, N$. For a projected classifier to be useful, it must be easy to compute the projection, and it must be effective in rejecting labelings at an early stage. These are strong requirements which are not satisfied by most good classifiers; for example, in our experience a support vector machine with a positive definite quadratic kernel projects easily but typically yields unrestrictive projected classifiers.

We have been using an axis-aligned bounding box, with bounds learned from a collection of positive labellings, for a good first separation, and then using a boosted version of a weak classifier that splits the feature space on a single feature value (as in [15]). This yields a classifier that projects particularly well,

and allows clean and efficient algorithms for computing projected classifiers and expanding sets of labels (see [23]).
The segment finder may find either 1 or 2 segments for each limb, depending on whether it is bent or straight; because the pruning is so effective, we can allow segments to be broken into two equal halves lengthwise, both of which are tested.

3.1 Results

The training set included 79 images without people, selected randomly from the COREL database, and 274 images each with a single person on uniform background. The images with people have been scanned from books of human models [41]. All segments in the test images were reported; in the control images, only segments whose interior corresponded to human skin in colour and texture were reported. Control images, both for the training and for the test set, were chosen so that all had at least 30% of their pixels similar to human skin in colour and texture. This gives a more realistic test of the system performance by excluding regions that are obviously not human, and reduces the number of segments in the control images to the same order of magnitude as those in the test images.
The models are all wearing either swim suits or no clothes, otherwise segment finding fails; it is an open problem to segment people wearing loose clothing. There is a wide variation in the poses of the training examples, although all body segments are visible. The sets of segments corresponding to people were then hand-labeled. Of the 274 images with people, segments for each body part were found in 193 images. The remaining 81 resulted in incomplete configurations, which could still be used for computing the bounding box used to obtain a first separation. Since we assume that if a configuration looks like a person then its mirror image would too, we double the number of body configurations by flipping each one about a vertical axis. The bounding box is then computed from the resulting 548 points in the feature space, without looking at the images without people.
The boosted classifier was trained to separate two classes: the $193 \times 2 = 386$ points corresponding to body configurations, and 60727 points that did not correspond to people but lay in the bounding box, obtained by using the bounding box classifier to incrementally build labelings for the images with no people. We added 1178 synthetic positive configurations obtained by randomly selecting each limb and the torso from one of the 386 real images of body configurations (which were rotated and scaled so the torso positions were the same in all of them) to give an effect of joining limbs and torsos from different images rather like childrens' flip-books. Remarkably, the boosted classifier classified each of the real data points correctly but misclassified 976 out of the 1178 synthetic configurations as negative; the synthetic examples were unexpectedly more similar to the negative examples than the real examples were.
The test dataset was separate from the training set and included 120 images with a person on a uniform background, and varying numbers of control images,

Features	# test images	# control images	False negatives	False positives
367	120	28	37%	4%
567	120	86	49%	10 %

Table 1. *Number of images of people and without people processed by the classifiers with 367 and 567 features, compared with false negative (images with a person where no body configuration was found) and false positive (images with no people where a person was detected) rates.*

reported in table 1. We report results for two classifiers, one using 567 features and the other using a subset of 367 of those features. Table 1 shows the false positive and false negative rates achieved for each of the two classifiers. By marking 51% of test images and only 10% of control images, the classifier using 567 features compares extremely favourably with that of [8], which marked 54% of test images and 38% of control images using hand-tuned tests to form groups of four segments. In 55 of the 59 images where there was a false negative, a segment corresponding to a body part was missed by the segment finder, meaning that the overall system performance significantly understates the classifier performance. There are few signs of overfitting, probably because the features are highly redundant. Using the larger set of features makes labelling faster (by a factor of about five), because more configurations are rejected earlier.

4 Shading primitives, shape representations and clothing

Finding clothed people is a far more subtle problem than finding naked people, because the variation in colour, texture and pattern of clothing defeats a colour segmentation strategy. Clothing does have distinctive properties: the patterns formed by folds on clothing appear to offer cues to the configuration of the person underneath (as any textbook on figure drawing will illustrate). These folds have quite distinctive shading patterns [19], which are a dominant feature of the shading field of a person clad in a loose garment, because, although they are geometrically small, the surface normal changes significantly at a fold. Folds are best analysed using the theory of buckling, and arise from a variety of causes including excess material, as in the case of a full skirt, and stresses on a garment caused by body configurations. Folds appear to be the single most distinctive, reliable and general visual cue to the configuration of a person dressed in a cotton garment.

4.1 Grouping folds using a simple buckling model

Garments can be modelled as elastic shells, allowing rather simple predictions of the pattern of folds using the Von Karman-Donnell equation or a linearised version of that equation. This is known to be a dubious source of predictions of buckling force, but the frequencies of the eigenfunctions — which give the buckled solutions — are accepted as fair predictions of the buckling mode for

the cases described (this is the topic of a huge literature, introduced in [5]). The eigenfunctions allow us to predict that garments buckling in compression or torsion will display long, nearly straight folds that are nearly parallel and nearly evenly spaced. These folds will be approximately perpendicular to the direction of compression and will indicate the direction of the torsion. The number of folds depends on tension in the garment, and is hard to predict.[3] For torsion, reasonable estimates of a garment's size yield on the order of five visible folds. As figures 4 and 5 indicate, these predictions are accurate enough to drive a segmentation process.

We apply the simple fold finder described in [20] to the image at twelve different orientations. Using these twelve response maps, we use non-maximum supression to find the centre of the fold, and follow this maximum along the direction of maximum response to link all points corresponding to a single fold. The linking process breaks sharp corners, by considering the primary direction of the preceding points along the fold.

After finding all of the folds in the image, the next step is to find pairs which are approximately parallel, and in the same part of the image. If the projections of the two folds onto their average direction are disjoint, they are considered to belong to different parts of the image.

From the theory, we expect that multiple folds will be at regularly spaced intervals. Thus, we look for pairs which have one common fold, and consistent separations. (The separations should either be the same, or one should be double the other—if a single fold gets dropped, we do not want to ignore the entire pattern.) The separation between folds is required to be less than the maximum length of the folds. Finally, some of these groups can be further combined, if the groups have almost the same set of folds.

The program typically extracts 10–25 groups of folds from an image. Figure 4 shows one image with three typical groups. The group in 4(b) clearly corresponds to the major folds across the torso in the image. This is in fact a segmentation of the image into coherent regions consisting of possible pieces of cloth. The region covered by the folds in (b) is most of the torso of the figure, and suggests a likely candidate for consideration as a torso. There are other groups as well, such as (c), the venetian blinds, and (d), an aliased version of (d), but these extra segments are easily dealt with by higher level processes.

Any image of man-made scenes will have a number of straight parallel lines which may have similar shading to folds (see, for example, figure 6). While this may be initially interpreted as groups of folds—hence as clothing—higher-level reasoning should enable us to reject these groups as coming from something other than folds in cloth.

[3] This can be demonstrated with a simple experiment. Wearing a loose but tucked-in T-shirt, bend forward at the waist; the shirt hangs in a single fold. Now pull the T-shirt taut against your abdomen and bend forward; many narrow folds form.

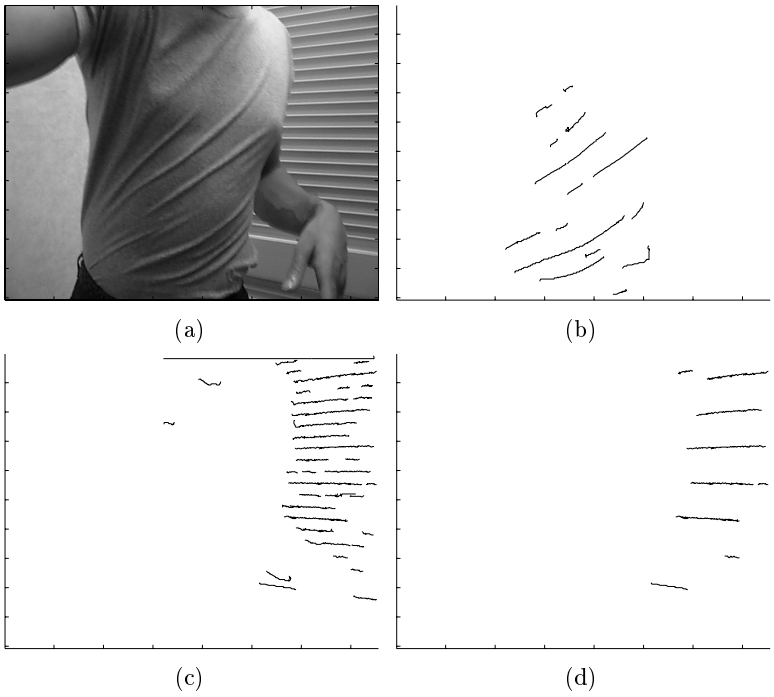


Fig. 4. *Results of a segmenter that obtains regions by grouping folds that satisfy the qualitative predictions of the linear buckling theory. (a) An image showing folds corresponding to torsional buckling. (b,c,d) Three groups of folds found by our program. The group in (b) is, in essence the torso; it contains the major folds across the torso, and can be used to represent the torso. An edge detector could not extract the outline points of the torso from this image, since the venetian blinds would result in a mess of edges. The group of fold responses in (c) is due to the venetian blinds in the background. Such a large set of parallel lines is unlikely to come from a picture of a torso, since it would require the torso to be unrealistically long. (d) A group that is an aliased version of the group in (c). Each group has quite high level semantics for segmenter output; in particular, groups represent image regions that could be clothing.*

4.2 Grouping folds by sampling

An alternative approach is to obtain groups which are samples from a posterior on groups given image data. This approach has the virtue that we do not need to come up with a detailed physical model of garment buckling — a process complicated by cloth anisotropy, etc. A simple likelihood model can be fitted to groups in real images, instead.

We describe each group of folds by a coordinate system and a series of variables which describe the scale of the folds, their angle, and their location with respect to a coordinate system. We also include the change in angle between adjacent folds (this enables us to describe star-shaped folds). By examining a number of

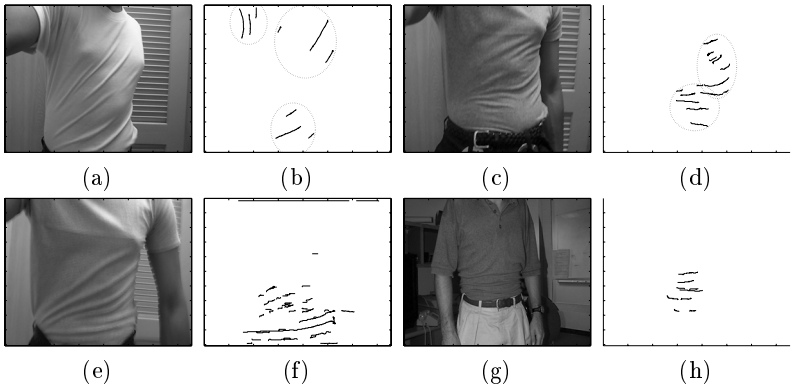


Fig. 5. *Further examples of segmentations produced by our grouping process. The figures show groups of fold responses, for the torsional (b,f) and axial (d,h) cases. In some cases, more than one group should be fused to get the final extent of the torso — these groups are separated by circles in the image. In each case, there are a series of between 10 and 25 other groups, representing either aliasing effects, the venetian blinds, or other accidental events. Each group could be a region of clothing; more high-level information is required to tell which is and which is not.*

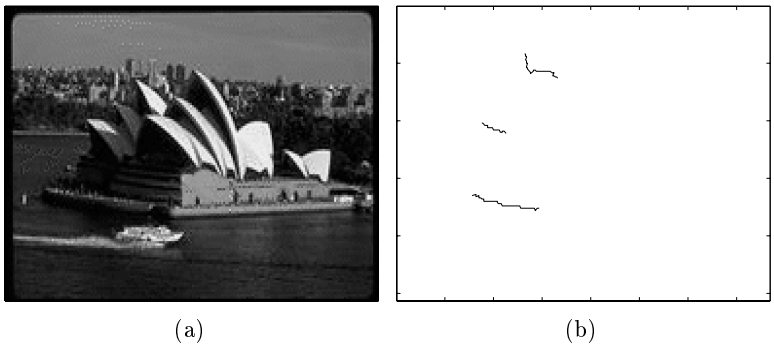
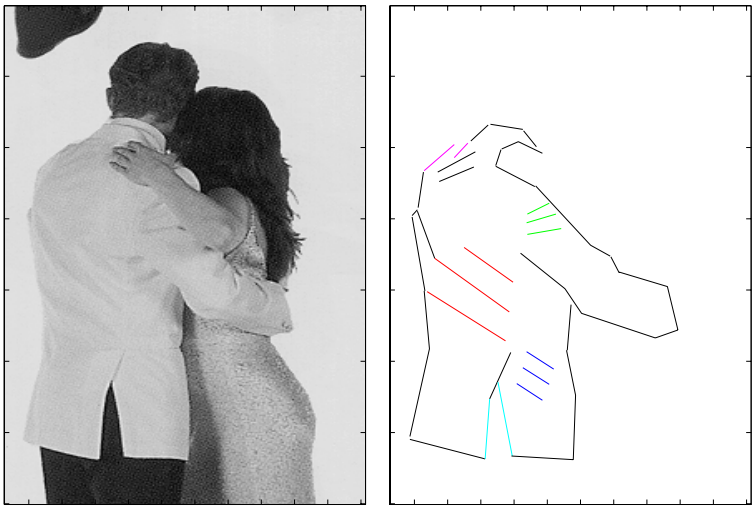


Fig. 6. *There are parallel folds that appear without clothing, too; (a) An image of an architectural curiosity. (b) One of four groups of folds found in the image. It is certainly expected that in images of man-made scenes, there will be a large number of nearly-parallel lines, which may be interpreted as groups of folds. Other cues should allow us to determine that this is not in fact clothing.*

groups in real images, we estimate a probability distribution on the parameters of the coordinate system. This allows us to describe how likely a group with those parameters is. We also estimate the probability distribution for individual folds within a group.



(a) (b)

Fig. 7. *The Markov Chain Monte Carlo method can be used to group folds together. (a) The original image. (b) Folds marked by hand, but grouped automatically. This is the most popular grouping of the image, after 10,000 iterations. Note that parallel folds are grouped together, and that the outline of the figure is largely ignored.*

The folds are grouped by running a reversible-jump Markov Chain Monte Carlo algorithm (as in [17]. If a fold has a high likelihood of belonging to a particular group, an assignment of the fold to that group should be fairly stable. In other words, it will have a high probability in the stationary distribution. The assignments which appear most frequently over a large number of iterations are taken to be the correct grouping. Proposal moves for this MCMC grouper are:

1. Add a new group. Two folds which have not previously been assigned to another group are combined to form a new group.
2. Delete a two-fold group.
3. Change the parameters of a group.
4. Add a fold to a group. An unassigned fold is assigned to an existing group.
5. Remove a fold from a group
6. Change the group of a fold. Change the group assignment of a fold.

After several thousand iterations, we observe that the MCMC spends a relatively high proportion of its time in certain states. We take the grouping in the most popular state to be the best grouping of folds for the images. Figure 7 shows an image, and the most popular grouping of folds. (The lower-level fold finder is not yet robust enough to generate reliable folds, so the putative folds here were marked by hand.) Parallel groups are taken to be a unit, and the edge of the figure is largely ignored, as desired.

4.3 Choosing primitives and building representations

Clothing is an interesting case because it is not obvious that folds are the right primitive to use. This raises the standard, difficult question that any theory based on primitives must address — how do we determine what is to be a primitive? As a possible alternative to our current fold-finder, we have been studying a mechanism for determining what should be a primitive following the ideas of [1, 2]. We obtain a large set of images of regions showing regions of folds, at the same orientation and scale. There is a comparison set, containing non folds that are not easy to distinguish from the folds using crude methods (e.g. a linear classifier on principal components). As measurements, we use spatial relations between filter outputs, for a reasonable set of filters at a variety of scales. We take uniform samples of subimages from each set.

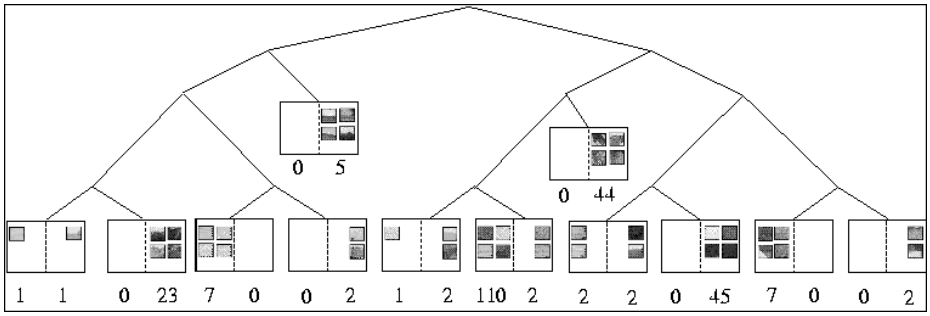


Fig. 8. *A representation of the decision tree used to find fold primitives. Each leaf contains a few windows representative of image windows classified at that level; on the left, clothing, and on the right, non-clothing. Below each leaf is the number of clothing and non-clothing windows that arrived at that leaf, out of a total of 128 in each category. 110 clothing and 2 non-clothing windows arrive at one leaf, strongly suggesting this combination of filter outputs is an appropriate clothing primitive.*

The task is now to explore the structure of the clothing set with respect to the non-clothing set. We do this by setting up a decision tree; each decision attempts to split the set at the leaf using an entropy criterion. The measurement used is the value of the output of one filter at one point — the choice of filter and point is given by the entropy criterion. The approach can be thought of as supervised learning of segmentation — we are training a decision tree to separate windows associated with objects to from those that are not. We split to several levels — a total of twelve leaves in the current experiments – and then use the representation at each leaf as a primitive. In particular, a leaf is defined by a series of filter outputs at a series of points; at each leaf we have an estimate of the frequency of observation of this pattern given clothing, and given no clothing. The remaining task is to postprocess the set of primitives to remove translational redundancies.

(a) (b)

Fig. 9. *Folds in clothing result from buckling and have quite characteristic shading and spatial properties, which are linked to the configuration of the person. (a) shows the probability that an image window centered at each point contains a clothing primitive, using automatically defined primitives sketched in figure 8; (b) shows lines of primitives linked together using an extremisation criterion. Note that edges are in general* not *marked, and that the process is insensitive to changes in albedo; these properties are a result of the learning process.*

5 Conclusions

For recognition systems to be practically useful, we need a system of representation that can handle a reasonable level of abstraction and that can support segmentation from quite general backgrounds. These requirements strongly suggest representations in terms of relations between primitives. We have shown that, using a simple primitive that is obviously convenient and useful, it is possible to build relational representations that are quite effective at finding naked people and horses. Furthermore, we have shown that a grouping process that finds such assemblies can be learned from data. These representations are crucially limited by the crude primitives used.
Primitives need not just be stylised shapes. The stylised appearance of folds in clothing means that we can study the appearance of clothing in a reasonably effective way. These are shading primitives. Although it is currently difficult to know how to choose primitives, the problem appears to be statistical on its face. Statistical criteria appear to be able to suggest promising choices of shading primitives from image data.

Acknowledgements

Thanks to Stuart Russell for focussing our attention on the attractions of MCMC as an inference method. The discussion of body plans is based on joint work with Margaret Fleck. Various components of this research were carried out with the support of an NSF Digital Library Grant (IRI 94-11334) and an NSF Graduate Fellowship to S.I.

References

1. Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9:1545–1588, 1997.
2. Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE T. Pattern Analysis and Machine Intelligence*, 19(11):1300–1305, 1997.
3. D.C. Blair. Stairs redux: thoughts on the stairs evaluation, ten years after. *J. American Soc. for Information Science*, 47(1):4–22, 1996.
4. D.C. Blair and M.E. Maron. An evaluation of retrieval effectiveness for a full text document retrieval system. *Comm. ACM*, 28(3):289–299, 1985.
5. C.R. Calladine. *Theory of shell structures.* Cambridge University Press, 1983.
6. P.G.B. Enser. Query analysis in a visual information retrieval context. *J. Document and Text Management*, 1(1):25–52, 1993.
7. O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *International Journal of Robotics Research*, 5(3):27–52, Fall 1986.
8. M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. In *European Conference on Computer Vision 1996, Vol. II*, pages 592–602, 1996.
9. M. Flickner, H. Sawhney, W. Niblack, and J. Ashley. Query by image and video content: the qbic system. *Computer*, 28(9):23–32, 1995.

10. D. A. Forsyth and M. M. Fleck. Identifying nude pictures. In *IEEE Workshop on Applications of Computer Vision 1996*, pages 103–108, 1996.
11. D.A. Forsyth and M.M. Fleck. Body plans. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.
12. D.A. Forsyth, M.M. Fleck, and C. Bregler. Finding naked people. In *European Conference on Computer Vision*, 1996.
13. D.A. Forsyth, J. Malik, M.M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. In *Proc. 2'nd International Workshop on Object Representation in Computer Vision*, 1996.
14. D.A. Forsyth, J.L. Mundy, A.P. Zisserman, C. Coelho, A. Heller, and C.A. Rothwell. Invariant descriptors for 3d object recognition and pose. *PAMI*, 13(10):971–991, 1991.
15. Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning - 13*, 1996.
16. M.M. Gorkani and R.W. Picard. Texture orientation for sorting photos "at a glance". In *Proceedings IAPR International Conference on Pattern Recognition*, pages 459–64, 1994.
17. P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.
18. W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Patt. Anal. Mach. Intell.*, 9(4):469–482, 1987.
19. J. Haddon and D.A. Forsyth. Shading primitives. In *Int. Conf. on Computer Vision*, 1997.
20. J. Haddon and D.A. Forsyth. Shape descriptions from shading primitives. In *European Conference on Computer Vision*, 1998.
21. A. Hampapur, A. Gupta, B. Horowitz, and Chiao-Fe Shu. Virage video engine. In *Storage and Retrieval for Image and Video Databases V – Proceedings of the SPIE*, volume 3022, pages 188–98, 1997.
22. D.P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. Int. Conf. Comp. Vision*, pages 102–111, London, U.K., June 1987.
23. S. Ioffe and D.A. Forsyth. Learning to find pictures of people. In *In review — NIPS*, 1998.
24. P. Lipson, W.E. L. Grimson, and P. Sinha. Configuration based scene classification and image indexing. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1007–13, 1997.
25. F. Liu and R.W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE T. Pattern Analysis and Machine Intelligence*, 18:722–33, 1996.
26. D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.
27. T.P. Minka and R.W. Picard. Interactive learning with a "society of models". *Pattern Recognition*, 30:465–481, 1997.
28. J.L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, Mass., 1992.
29. J.L. Mundy, A. Zisserman, and D. Forsyth. *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.
30. V.E. Ogle and M. Stonebraker. Chabot: retrieval from a relational database of images. *Computer*, 28:40–8, 1995.

31. R.W. Picard, T. Kabir, and F. Liu. Real-time recognition with the entire brodatz texture database. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 638–9, 1993.
32. J. Ponce. Straight homogeneous generalized cylinders: differential geometry and uniqueness results. *Int. J. of Comp. Vision*, 4(1):79–100, 1990.
33. J. Ponce, D. Chelberg, and W. Mann. Invariant properties of straight homogeneous generalized cylinders and their contours. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(9):951–966, September 1989.
34. L.G. Roberts. Machine perception of three-dimensional solids. In J.T. Tippett et al., editor, *Optical and Electro-Optical Information Processing*, pages 159–197. MIT Press, Cambridge, 1965.
35. G. Salton. Another look at automatic text retrieval systems. *Comm. ACM*, 29(7):n649–657, 1986.
36. S. Santini and R. Jain. Similarity queries in image databases. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 646–651, 1996.
37. M. Stricker and M.J. Swain. The capacity of color histogram indexing. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 704–8, 1994.
38. M.J. Swain. Interactive indexing into image databases. In *Storage and Retrieval for Image and Video Databases – Proceedings of the SPIE*, volume 1908, pages 95–103, 1993.
39. M.J. Swain and D.H. Ballard. Color indexing. *Int. J. Computer Vision*, 7(1):11–32, 1991.
40. D.W. Thompson and J.L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *IEEE Int. Conf. on Robotics and Automation*, pages 208–220, Raleigh, NC, April 1987.
41. unknown. *Pose file*, volume 1-7. Books Nippan, 1993-1996. A collection of photographs of human models, annotated in Japanese.
42. D.A. White and R. Jain. Imagegrep: fast visual pattern matching in image databases. In *Storage and Retrieval for Image and Video Databases V – Proceedings of the SPIE*, volume 3022, pages 96–107, 1997.
43. A. Zisserman, J.L. Mundy, D.A. Forsyth, J.S. Liu, N. Pillow, C.A. Rothwell, and S. Utcke. Class-based grouping in perspective images. In *Int. Conf. on Computer Vision*, 1995.