Finding people by sampling

Sergey Ioffe
Computer Science Division
U.C. Berkeley
Berkeley, CA 94720
ioffe@cs.berkeley.edu

David Forsyth
Computer Science Division
U.C. Berkeley
Berkeley, CA 94720
daf@cs.berkeley.edu

Abstract

We show how to use a sampling method to find sparsely clad people in static images. People are modeled as an assembly of nine cylindrical segments. Segments are found using an EM algorithm, and then assembled into hypotheses incrementally, using a learned likelihood model. Each assembly step passes on a set of samples of its likelihood to the next; this yields effective pruning of the space of hypotheses. The collection of available nine-segment hypotheses is then represented by a set of equivalence classes, which yield an efficient pruning process. The posterior for the number of people is obtained from the class representatives. People are counted quite accurately in images of real scenes using an MAP estimate. We show the method allows top-down as well as bottom up reasoning. While the method can be overwhelmed by very large numbers of segments, we show that this problem can be avoided by quite simple pruning steps.
Keywords: *Object recognition, sampling, Probabilistic inference*

1. Introduction

Finding people in static images is difficult, because the number of internal degrees of freedom defeats simple correspondence reasoning. However, people can be quite accurately modeled as assemblies of cylinders, and these assemblies are constrained by the kinematics of human joints. There is a long tradition of using these constraints to find people (e.g. [1, 6, 8, 3]; pedestrians in a standard configuration can be found by template matching [7]). No existing work can count people, and serious difficulties with segmentation remain.

These segmentation difficulties can only be overcome by using object level knowledge as early as possible in the segmentation process. We represent people as collections of nine body segments, one for the torso and two for each limb (the face could be dealt with by current, very accurate, face-finding algorithms [10, 9]). In this strategy, we find individual body segments; these segments are then assembled into pairs that satisfy kinematic constraints; the pairs are assembled into triples, etc. The main advantage of this approach is that poor hypotheses can be pruned early (as in [5]). However, there is the danger of pruning a hypothesis that is locally poor but which is a component of a good global hypothesis. This is a common problem in recognition — false negatives are much harder to resolve than false positives — and is aversion of the horizon problem in search.

We finesse this difficulty by using a probabilistic inference method. A standard method forms a posterior, and then represents possible inferences by drawing samples from this posterior [4]. Building a good sampler for finding people is tricky, because the posterior that a person is present given a single segment will be very small, so that it is difficult to start the assembly process. Instead, we *sample the likelihood.* We use the term "assembly" to refer to a group of segments, labeled with correspondence to human body segments. For any nine segment assembly A, define the likelihood $L(A) = \Pr[A \text{ will appear in the image}|\text{a person is present}]$. We now sample subassemblies from the available segments in the image according to marginalised versions of this distribution. This prunes the set of assemblies without denying any hypothesis a chance to grow. We show the results may be used to count people in the image, segment them from the background, and infer their configurations, and find body parts missed by the original segmentation.

1.1. Resampling

There are too many nine segment assemblies to compute the likelihood for each. However, we can build assemblies incrementally. For example, having generated a set of samples $\{s_T\}$ of potential `torso` segments and samples $\{s_{LUA}\}$ of `left upper arms`, we can form all combinations $\{s_T, s_{LUA}\}$ and then resample it, so that the resulting pairs (s_T, s_{LUA}) come from the appropriate marginal like-

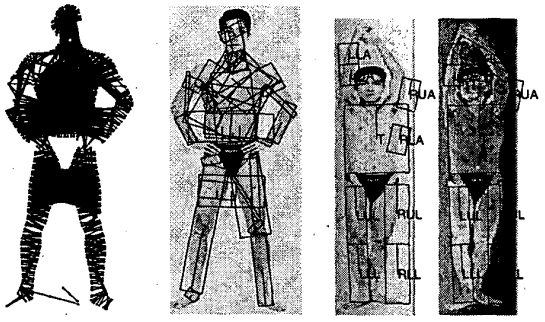


Figure 1. *Symmetries* (a) *and segments* (b) *produced for an image.* (c): *Assemblies corresponding to the same person often share* `torsos`

lihood. We can proceed by similarly sampling 3-, 4-, ..., 9-segment sub-assemblies, in such a way that the resulting set of 9-segment assemblies is sampled from $L(\cdot)$.

At each stage, we use *importance sampling*, which is a method for drawing samples from (possibly intractable) distributions (as used in [2]). In particular, to draw a sample from $g(x)$, we first draw a large number of independent samples $\{s_1, \ldots, s_n\}$ from a proposal distribution $f(x)$, and then set $s = s_i$ with probability proportional to $w_i = \frac{g(x)}{f(x)}$. As $n \to \infty$, the distribution for the sample s will approach $g(x)$. In our case, the proposal distributions are the marginal likelihoods for the subassemblies. Thus, we are more likely to propose a pair (s_{RUA}, s_{RLA}) if the two segments individually are more likely to be `upper right arm` and `lower right arm` of a person.

2. Implementation

Our system starts by finding *symmetries* (fig. 1(a)), which are pairs of edge elements that are approximately symmetric about some symmetry axis and whose tangents are approximately parallel to that axis. These symmetries are grouped into *segments* — extended groups of symmetries which approximately share the same axis — (fig. 1(b)) using an expectation-maximization algorithm that assumes a fixed number of segments. From the segments, we use a learned *likelihood model* to form *assemblies* by sampling (fig. 1(c)). Finally, the set of assemblies is replaced with a smaller set of *representatives*, which are used to count people in the image.

2.1. Finding Segments Using EM

Each segment is represented with a *symmetry axis* and a *width*. Each symmetry has a label showing which of at most one segment it belongs to. A symmetry fits a segment best when the midpoint of the symmetry lies on the segment's symmetry axis, the endpoints lie half a segment width away from the axis, and the symmetry is perpendicular to the axis (that is, the axes of symmetry of the symmetry and the segment coincide). This yields the conditional likelihood for a symmetry given a segment as a four-dimensional Gaussian (two numbers for each endpoint), and an EM algorithm can now fit a fixed number of segments to the symmetries. After that, we determine where each segment begins and ends by finding the range of symmetries for which this segment has the largest posterior. If there is a large gap between these symmetries (that is, symmetries from different image regions are attributed to the same segment), then the segment is broken into two or more pieces.

2.2. Representing Likelihoods for People

The likelihood for a nine segment assembly is computed from a set of 41 geometric features, invariant to translation, rotation and scale. These include angles and distances between segments, aspect ratios of segments, length ratios, etc. As nine rectangles have 41 degrees of freedom up to a rigid transformation, we choose the features so as to have a one-to-one correspondence between the feature space and the space of all assemblies. Each feature in our model depends on either one or two segments, and the two-segment features can be computed either from the two halves of the same limb (such as `right upper arm` and `right lower arm`), or from an upper limb and the torso.

This choice of features allows us to assume that features are independent with a relatively small error. The main errors will be due to interactions between kinematic constraints on the hips and shoulders, and viewing pose. This assumption is attractive because the likelihood has an especially simple form,

$$L(A) = \prod_{i=1}^{41} d_i(f_i), \qquad (1)$$

where f_i is the value of the ith feature, and $d_i(f_i)$ is the corresponding one-dimensional marginal likelihood. In our experiments, we chose for $d_i(\cdot)$ to be a histogram for the values f_i.

2.3. Building Assemblies Incrementally by Resampling

We fix a permutation $(l_1, \ldots, l_9)$ of labels $\{\mathrm{T}, \mathrm{LUA}, \ldots\}$, and generate a sequence $(S_1, \ldots, S_9)$

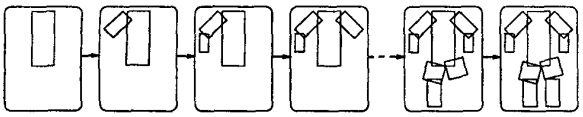


Figure 2. *We sample assemblies incrementally, by generating sets of samples of 1-, 2-, ..., 9-segment assemblies, so that the latter are drawn from the likelihood $L(\cdot)$*

of multisets of samples, where each S_k contains N (not necessarily distinct) assemblies of k segments labeled as $l_1, \ldots, l_k$ (fig. 2). For example, in our implementation, $(l_1 \ldots l_9) = (\mathrm{T}, LUA, \mathrm{L}LA, \ldots)$, and so S_1 will contain the samples (s_T) of `torso` segments, while S_3 will contain samples $(s_\mathrm{T}, s_{LUA}, s_{LLA})$ of triples corresponding to the `torso`, the `left upper arm` and the `left lower arm`. The samples in S_k are drawn from the *marginal likelihood* $L_{l_1 \ldots l_k}(A) = \prod_i d_i(f_i)$, where the product is over all the features computable from segments labeled as $l_1, \ldots, l_k$. We write s_{l_i} for the segment of the sub-assembly whose label is l_i. For our feature set and the choice of $(l_1 \ldots l_9)$, each of the marginal likelihoods $L_{l_1 \ldots l_k}(s_{l_1}, \ldots, s_{l_k})$ models the probability that the sub-assembly $(s_{l_1}, \ldots, s_{l_k})$ is seen in a random view of a human.

We generate the set of samples S_{k+1} from S_k using importance sampling. First, we form the set of sub-assemblies $(s_{l_1}, \ldots, s_{l_k}, s_{l_{k+1}})$ for all groups $(s_{l_1}, \ldots, s_{l_k}) \in S_k$ and all choices of $s_{l_{k+1}}$. The first component is a sample from the relevant marginal distribution. We now *resample* this set of samples, by independently drawing N samples, with the probability of drawing $(s_{l_1}, \ldots, s_{l_{k+1}})$ proportional to $w(s_{l_1}, \ldots, s_{l_{k+1}}) = \frac{L_{l_1 \ldots l_{k+1}}(\cdot)}{L_{l_1 \ldots l_k}(\cdot)} = \prod_i d_i(f_i)$, where the product is over all features that depend on $s_{l_{k+1}}$ and, possibly, some of $s_{l_1}, \ldots, s_{l_k}$.

2.4. Directing the sampler

Our sampler is working in a discrete space of labels and image segments. It can be difficult to focus the activity of such samplers on components with large probability. For example, if there are two people in the image, and one results in a large group of segments and the other in a small group (due to mischief in the segment finder), the sampler may repeatedly draw samples from the large group corresponding to the one person, and never get to the other. A natural strategy is to break the domain into a set of equivalence classes, sample the classes, and then sample within the classes drawn by that sampler.

We define equivalent assemblies to be those that label the same segment as a `torso`. This is a good choice, because different people in an image will tend to have their torsos in different places. We represent the class by the assembly that has the highest likelihood. This means that we have a tight upper bound for the likelihoods within the equivalence class, which means that classes that are omitted when we sample classes tend to be those which contain elements of relatively low likelihood. For an exact algorithm we would need elements within classes to have similar likelihoods; our results suggest that this is not particularly important.

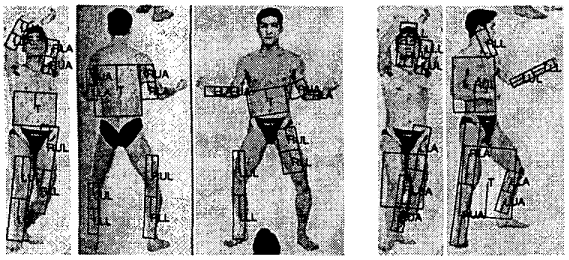
The highest likelihood assembly is found by a simple greedy algorithm. As an example, suppose that all of the segments in an assembly, except the lower left arm, are fixed, and we are to choose the lower left arm that maximizes the likelihood of the resulting assembly. It is easy to see that, in our model, the lower left arm can be found by considering all the pairs of a lower left arm (which can be any segment) and the upper left arm (which is fixed), and choosing the one with the highest marginal likelihood $L_{LUA,LLA}$. Now, let us suppose that we have fixed a `torso` and, possibly, some limbs, and we want to add the left arm that would maximize the likelihood of the result. First, we will find the highest-likelihood left arm for each choice of the upper arm. Since no feature involves the left arm and any other limb, we can choose the best left arm by considering all the pairs of the `torso` (which is fixed) and a left arm, and choosing the one with the largest marginal likelihood.

Now we have a greedy algorithm which, for each choice of upper left arm, finds the lower left arm so as to maximize the marginal $L_{LUA,LLA}$, and similarly for the other limbs. Then, for each possible `torso` segment, the limbs are added in a sequence, maximizing the corresponding marginal likelihoods $L_{\mathrm{T},LUA,LLA}, L_{\mathrm{T},LUA,LLA,RUA,RLA}$, etc. At the end, we have the largest likelihood assembly for each `torso` segment. The algorithm is efficient: if there are n segments in the image, we never have more than n sub-assemblies of each type, thus the algorithm runs in $O(n^2)$ time (and much faster in practice, if we only try to pair up segments that are close).

Although the upper bounds provided by this algorithm are very effective for directing the sampler to relevant image regions, they may not be tight. For example, in the resulting assemblies the legs may coincide, since ensuring distinct legs would require a (binary) feature involving both legs. Such assemblies do not consist of 9 segments; they do, however provide upper bounds on the likelihoods of assemblies with a given segment as the `torso`.

3. Counting People

Our sampling algorithm allows to count people in images. To estimate the number of people, we begin by select-



a b

Figure 3. *The representatives that do* (a) *and do not* (b) *correspond to the configurations of people in the images.*

ing a small set of *representative assemblies* in the image, and then use them for counting.

3.1. Finding representative assemblies

We assume that distinct people have distinct torsos, accepting that occlusion of one torso by another will lead to a miscount. We break the set of all assemblies in the image into (not necessarily disjoint) *blocks* — sets of assemblies such that any two assemblies from the same block have overlapping `torsos`. Then, the *representative* is chosen from each block as the assembly with the highest likelihood, over all assemblies available from the block. Because we have assumed that any people in the image are spaced apart, we can use representatives to count people — by replacing the set of assemblies with that of representatives, we do not diminish the count. Indeed, any assembly that is not a representative must be overlapped by a higher-likelihood representative, and so if there was a human assembly in some region of the image, there will be a representative there as well. In fact, the configuration of a person can often be inferred from that of representatives (Fig. 3).

We can efficiently find representatives, since we can use the upper bounds on the likelihoods, computed in Section 2.4. In particular, if the algorithm of Sec. 2.4 produced a valid assembly (no coinciding segments) for some `torso` segment, then sampling need not be performed for that `torso` (since this assembly has a higher likelihood than any other we can obtain by sampling). If, however, the assembly obtained for the upper bound is not a valid one, we have to sample assemblies with the given `torso`, but only retain the one with the highest likelihood (since all of the assemblies share the `torso`). Furthermore, we need not sample for a given `torso` segment if there is already an overlapping assembly, whose likelihood is greater than the upper bound for the given `torso`.

3.2. Estimating the number of people

Once the representative set has been computed for an image, we want to obtain the estimate on the number of people in the image. We assume that assemblies corresponding to people do not overlap and have independent configurations. Let the set of representatives be $\{A_1, \ldots, A_m\}$, and let us consider any set $G \subseteq \{1 \ldots m\}$, such that no assemblies from $\{A_i | i \in G\}$ overlap. We will look at the *posterior probability* Pr[each of A_i represents a person|image data] that the representatives $\{A_i | i \in G\}$ are people while $\{A_j | j \notin G\}$ are not. To count people, we choose the set G for which the posterior is largest, and the size $|G|$ will give the MAP estimate of the number of people in the image. We could also represent this posterior as a set of samples to give some insight into the reliability of a particular count.

We assume that each assembly has the *a priori* probability β of being a person, independently of the others. Then, the prior for G is $\pi(G) = \beta^{|G|}(1-\beta)^{m-|G|}$, and the posterior is proportional to$\Pr[A_1, \ldots, A_m|G]\pi(G)$. Since the human assemblies do not overlap, $\Pr[A_1, \ldots, A_m|G] = 0$ if some of $\{A_i | i \in G\}$ overlap. Otherwise, we have $\Pr[A_1, \ldots, A_m|G] = \prod_{i \in G} L(A_i) \prod_{i \notin G} L_{non}(A_i)$, where we still use $L(A) = \Pr$[person in random configuration looks like A], and define $L_{non}(A) = \Pr[A|$random view not containing a person$]$. Finally, we assume $L_{non}(\cdot)$ to be uniform. We get that, for non-overlapping $\{A_i | i \in G\}$, the posterior is proportional to ${L_{non}}^{m-|G|} \prod_{i \in G} L(A_i) \beta^{|G|}(1-\beta)^{m-|G|}$, or

$$c^{|G|} \prod_{i \in G} L(A_i), \tag{2}$$

where the constant $c = \frac{\beta}{(1-\beta)L_{non}}$ is to be estimated so as to yield best classification.

4. Results

To learn the likelihood model $L(\cdot)$, we used a set of 193 training images, scanned from [11]. Each contained a photograph of a single person, standing against a uniform background. All the views were frontal and all limbs were visible, although the configurations varied. The models wore swimsuits or no clothes, since clothes make it hard to propose body segments. The symmetries produced for each image were used to determine sets of segments, although the segment finder was not the EM-based one used on the test data. We hand-labeled the segments by marking those corresponding to the 9 body segments. In fact, the training images were the part of a larger collection that resulted in complete assemblies (no segment finder misses). Since

the likelihood should not favor an assembly over its mirror image, we expanded the training set by adding the mirror image of each assembly, thus resulting in 386 configurations. The likelihood $L(\cdot)$ was defined as in Eqn. (1), where $d_i(\cdot)$ were the histograms (with 20 bins) for each of the 41 geometric features for the training set.

4.1. Test data

The test data included 145 *control images* with no people, and 228, 72, and 65 images with 1, 2, and 3 people, respectively. The control images came from the COREL database, while those with people were obtained by combining single-person images from the same collection as, but distinct from, the training data.

The sets of symmetries were produced for each test image. The parts of the control images differing significantly in color from people's skin (no more than 1/2 of each image) were blanked out before finding symmetries; no such preprocessing was done for images with people. The EM-based segment finder was applied to each set of symmetries by fitting 50 mixture components to each control image, 20 and 40 (on separate runs) to the 1-person images, and 40 and 60 to both 2- and 3-person images. The actual number of segments produced varied, due to splitting of segments with gaps. The resulting collections of segments were then used for testing.

To be able to find both straight (1 segment) and bent (2 segments) limbs, we added both halves (lengthwise) of each segment to the segment sets. The halves of a segment, however, could only appear either together in the same limb, or as the `torso`.

4.2. People vs No people

We used sampling and representative selection to count people, as in Sec. 3.2. For each image, we found the MAP subset $\{A_i | i \in G\}$ of representatives classified as people, and classify the image as containing a person if $|G| \geq 1$, and no people if $G = \emptyset$. Fig. 4(a) shows how the success of this classification depends on the value of c, from Eqn. (2).

4.3. Counting people

Similarly to the above, we used the size $|G|$ of the MAP set G as the estimate of the number of people. Fig. 4(b) shows, for images with $k = 0 \ldots 3$ people, the fraction of segment sets that yielded the correct estimate $|G| = k$.

The 3-person images did not yield as good results as those with fewer people. This could be due either to the fact that with more people in the image the segment finder is more likely to miss a body segment, or to our choice of

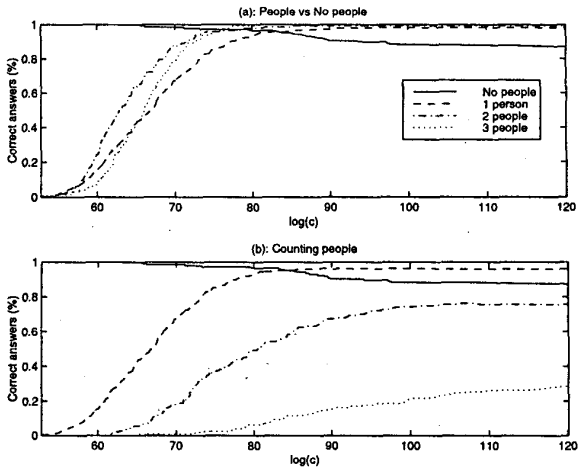


Figure 4. *Percentage of correct decisions for Person vs No person classification (a) and Counting (b), as a function of c. Each figure shows the percentages separately for images with 0,1,2, and 3 people*

representatives: it is possible that, while non-overlapping assemblies exist for each of the people in the image, the representatives do overlap, thus diminishing the people count. For many cases, the representatives give quite a good indication of the configuration of the people present (figure 6).

5. Discussion

The control set used in these results had been censored to remove regions of high texture and of a particular range of colours (censored regions in figure 5 are shown in white). This significantly reduces the number of segments reported. If one uses an uncensored control set, the program almost always finds one person because the number of available segments overwhelms the selectivity of our constraints. This suggests that segment finding is insufficient to segment people; other possible tests include using the characteristic contour shape of muscle or a more detailed shading test.

Seeing recognition as an inference problem has the advantage that top-down information flow can coexist with bottom up information flow quite reasonably. Often, the segments corresponding to one or more of a person's body parts are missing from the segment set of the image. This can be caused by either occlusion or a failure of the segment finder. For such *incomplete assemblies*, the likelihood $L(\cdot)$ is not available; nevertheless, we want to be able to find incomplete assemblies. Furthermore, having found one, we want to guess where the missing segments could be. Then, we could go back to the image and try and analyze the possi-

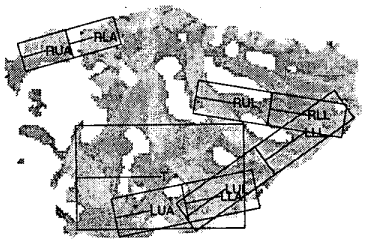


Figure 5. *A control image for which a human assembly was found*

bility of the occlusion, or re-run the segment finder, paying closer attention to the specified image regions, so as to find the missing segments. A method currently in development would solve the problem by first adding a large number of random "dummy segments" to the segment set, and then running our original sampling algorithm, limiting the number of dummy segments in an assembly. This would allow to obtain samples of incomplete assemblies from the corresponding marginal likelihoods, and those of missing segments (the dummy segments in assemblies) — from conditionals Pr[missing segments$|A_{inc}$], for each incomplete assembly A_{inc}.

Performance of our algorithm would be improved by a better likelihood model and by principled feature selection. Future work will involve incorporating the segment finder and the assembly builder in a single Markov Chain Monte-Carlo framework yielding a chain of probabilistic reasoning from pixel to person.

Acknowledgements

SI is supported by an NSF Graduate Fellowship. Thanks to Stuart Russell for pointing out the significance of MCMC as an inference technique.

References

[1] G. Agin. *Representation and description of curved objects.* PhD thesis, Stanford University, Stanford, CA, 1972.

[2] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion.* Springer Verlag, 1998.

[3] D. Forsyth, M. Fleck, and C. Bregler. Finding naked people. In *European Conference on Computer Vision*, 1996.

[4] W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov chain Monte Carlo in practice.* Chapman and Hall, 1996.

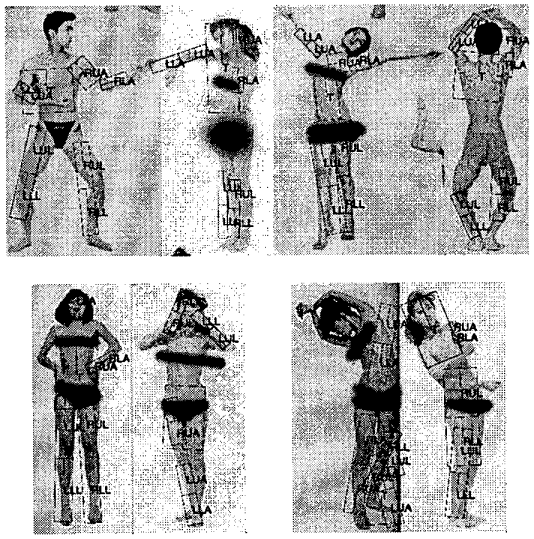


Figure 6. *Examples showing representatives for images with two people; these representatives give quite a good guide to the person's configuration (top row); the bottom row shows bad cases. Images have been airbrushed so they can be shown* salve pudore.

[5] W. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3), 1984.

[6] R. Nevatia and T. Binford. Description and recognition of complex curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[7] M. Oren, C. Papageorgiou, P. Sinha, and E. Osuna. Pedestrian detection using wavelet templates. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 193–9, 1997.

[8] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE T. Pattern Analysis and Machine Intelligence*, 2:522–546, 1980.

[9] T. Poggio and K.-K. Sung. Finding human faces with a gaussian mixture distribution-based face model. In *Asian Conf. on Computer Vision*, pages 435–440, 1995.

[10] H. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing 8*, pages 875–881, 1996.

[11] E. Shuppan. *Pose file*, volume 1-7. Books Nippan, 1993-1996. A collection of photographs of human models, annotated in Japanese.