Stenger, B, Thayananthan, A, Torr, P H and Cipolla, R
Estimating 3D hand pose using hierarchical multi-label classification.

Stenger, B, Thayananthan, A, Torr, P H and Cipolla, R (2007) Estimating 3D hand pose using hierarchical multi-label classification. Image and Vision Computing , 25 (12). pp. 1885-1894.

doi: 10.1016/j.imavis.2005.12.018

Estimating 3D Hand Pose Using Hierarchical Multi-Label Classification

Björn Stenger [*], Arasanathan Thayananthan [†], Philip H. S. Torr [‡],
and Roberto Cipolla [§]

Abstract

This paper presents an analysis of the design of classifiers for use in a hierarchical object recognition approach. In this approach, a cascade of classifiers is arranged in a tree in order to recognize multiple object classes. We are interested in the problem of recognizing multiple patterns as it is closely related to the problem of locating an articulated object. Each different pattern class corresponds to the hand in a different pose, or set of poses. For this problem obtaining labelled training data of the hand in a given pose can be problematic. Given a parametric 3D model, generating training data in the form of example images is cheap, and we demonstrate that it can be used to design classifiers almost as good as those trained using non-synthetic data. We compare a variety of different template-based classifiers and discuss their merits.

1 Introduction

This paper considers the problem of locating and tracking an articulated object using a single camera. The method is illustrated by the problem of hand detection and tracking. There is a lot of ambiguity in the problem of tracking a complex articulated object, and a successful method should be able to maintain multi-modal distributions over time. A number of different techniques have been suggested to deal with multi-modality, e.g. particle filtering [9, 13] or grid based methods [24, 25]. When track is lost, a robust tracker should devise a recovery strategy, as for example in region based tracking [28]. This task, however, can be seen as a detection problem, and thus in [24, 25] it is argued that the tracking of complex objects should involve the close synthesis of object detection and tracking.

Object recognition is typically considered as the task of detecting a single class of objects O (e.g. faces) in a scene I; locating the object in the scene and determining its pose. But suppose we are interested in recognizing m categories of objects $O_1, \ldots, O_m$ simultaneously, i.e. are any of a set of objects in the scene, and if so where? How can it efficiently be decided whether the scene contains one of these objects? One option that is commonly followed is to independently train a classifier for each object [20]. The drawback of such an approach is that computation time scales roughly linearly in the number of objects to be identified. Baker and Nayar used low-cost classifiers with high detection rate and moderate false positive rate, which they called *rejectors*, in an object recognition application [2]. A hierarchy of such classifiers was built, reducing the computational cost to be logarithmic in the number of classes. Recently advances have been made in face detection based on the idea of a cascade of classifiers [19, 27], where successively more complex classifiers are combined in a cascade structure, which increases the speed of the detector by focusing attention on promising regions of the image, see figure 1(a). First the image is divided into a set of subregions. The initial classifier eliminates a large portion of these subwindows with little computation; those remaining are processed further down

[*] B. Stenger is with the Toshiba Cambridge Research Laboratory, Cambridge, CB2 3NH, UK. E-mail: bjorn@cantab.net.

[†] A. Thayananthan is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK. E-mail: at315@cam.ac.uk.

[‡] P.H.S. Torr is with the Department of Computing, Oxford Brookes University, Oxford OX33 1HX, UK. E-mail: philiptorr@brookes.ac.uk.

[§] R. Cipolla is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK. E-mail: cipolla@eng.cam.ac.uk.

the cascade. At each level the number of subwindows remaining decreases, allowing for more computationally expensive classifiers to be used at the bottom level for accurate discrimination of the remaining subwindows. As the motivation for such a cascade is the minimization of computation time, this paper examines some of the issues involved in classifier design for efficient template-based classification for hand pose estimation.

The next section reviews related work on hierarchical detection and describes the links to 3D pose estimation. Section 3 introduces the shape and colour features that are used, as well as the templates for classification. An evaluation of these classifiers in terms of performance and efficiency is presented in section 4, and their application to detection and pose estimation is demonstrated in section 5.

2 Pose estimation using shape templates

One question is whether a cascaded approach can be used for recognizing multiple objects, i.e. how to design a computationally efficient cascade of classifiers for a given set of objects, $\mathbf{O}_1, \dots, \mathbf{O}_m$. The problem is closely linked to the recognition of articulated objects which can be thought of as an infinite collection of objects indexed by the joint articulation parameters. In particular, Gavrila [8] examines the problem of detecting pedestrians. Chamfer matching [3] is used to detect humans in different poses, and detecting people is formulated as a template matching problem. When matching many similar templates to an image, a significant speed-up can be achieved by forming a template hierarchy and using a coarse to fine search [8, 18]. An approach to embed exemplar-based matching in a probabilistic tracking framework was proposed for complete image frames by Jojic and Frey [15] and for exemplar templates by Toyama and Blake [26]. The main idea is to use a discrete state model together with continuous transformation parameters. In [26] exemplar templates are used to evaluate likelihoods within a probabilistic tracking framework. Shape templates of a walking person are clustered and only the chamfer cost of the prototypes needs to be computed. However, with increasing object complexity the number of exemplars required for tracking rises as well. If a parametric 3D object model is available, the generation of training examples is cheap. Additionally, each generated 2D template is annotated with the 3D model parameters, thus pose recovery can be formulated as object detection: create a database of model-generated images and use a nearest-neighbour search to find the best match. This approach is followed, for example, by Athitsos and Sclaroff for hand pose estimation [1] and Shakhnarovich et al. [21] for upper body pose estimation.

In [24] it is suggested to partition the parameter space of a 3D hand model using a multi-resolution grid. A distribution is defined on the finest grid and is propagated over time. This has the advantage that temporal information can be used to resolve ambiguous situations and to smooth the motion. Shape templates, generated by the 3D model, are used to evaluate the likelihoods in regions of the state space. The templates are arranged in a hierarchy and are used to rapidly discard regions with low probability mass. For the first frame, the tree corresponds to a detection tree, thus the idea of cascaded classifiers can be applied, which eliminate large regions of the parameter space at early stages and focus computation on ambiguous regions. In terms of classifiers, the aim is to maintain a high detection rate while rejecting as many false positives as possible at each node in the tree. Within this paper we will analyze the design of cascaded classifiers for such a hierarchy, which can be used within the tree-based filtering framework of [24].

Given a tree which at each level partitions the set of models into mutually exclusive regions S_i^l for $l = 1, \dots, L$ where L is the number of levels in the tree and $i = 1 \dots N_L$ where N_L is the number of sets at that level. So that $S = \cup_i S_i^l$ and $S_j^l \cap S_k^l = 0; \forall j, k$. The goal is to design a classifier C_j^l which achieves high detection rates with modest false positive rates or the region S_j^l. The search then proceeds as shown in algorithm 1. A schematic of this algorithm is shown in figure 1b. The next section examines a number of different classifiers based on edge and colour features.

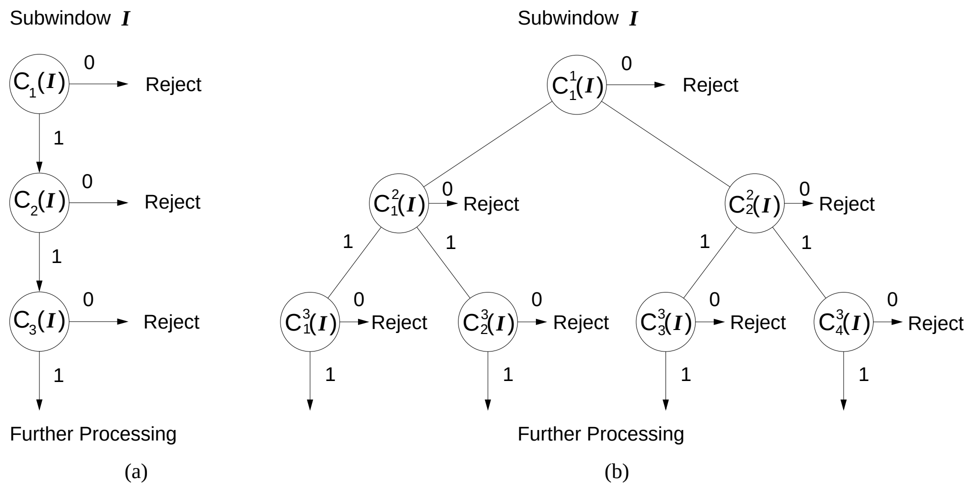


Figure 1: **Cascade of Classifiers**. **(a)** *A cascade of classifiers for a single object class where each classifier has a high detection and moderate false positive rate.* **(b)** *Classifiers in a tree structure; in a tree-based object recognition scheme each leaf corresponds to a single object class. When objects in the subtrees have similar appearance, classifiers can be used to quickly prune the search. A binary tree is shown here, but the branching factor can be larger than two.*

Algorithm 1 : Cascade of classifiers for multiple categories

```
for each subwindow $\mathbf{I}$
    start at root node $(l, k) = (1, 1)$.
    if $C_k^l(\mathbf{I}) > 0$ then repeat for child nodes of $(l, k)$.
    else assign zero probability to all child nodes of $(l, k)$.
    endif
endfor
```

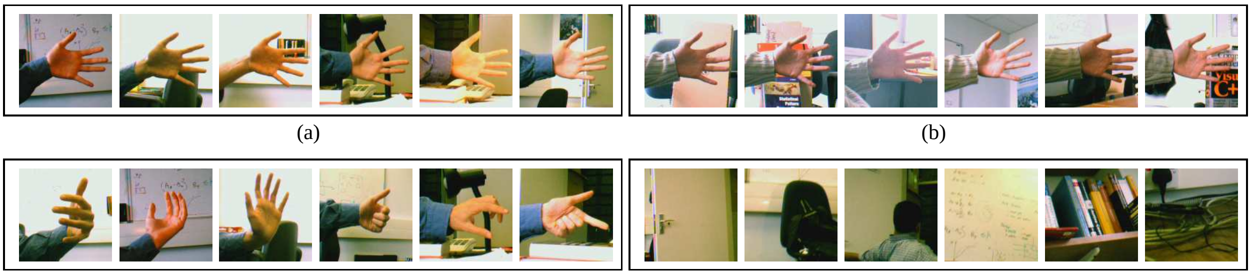
3 Explanation of features and classifiers

Edges (occluding contours) and colour (silhouette interior) have proved useful features for recognizing hands and discriminating between different poses, e.g. [1, 17]. Each feature is treated independently, assuming that in different settings one of the features may be sufficient for recognition. Edge features are considered in the following section.

3.1 Edge features

When using edges as features, robust similarity functions need to be used when comparing a template with the image, i.e. ones that are tolerant to small shape changes. One way to achieve this is to blur the edge image or template before correlating them. Other methods, which are tolerant to small shape deformations and some occlusion are the (truncated) chamfer and Hausdorff distance functions [3, 12]. Both methods are made efficient by the use of fast operations like the distance transform or dilation of the edge image. Olson and Huttenlocher [18] include edge orientation in Hausdorff matching. This is done by decomposing both template and edge image into a number of separate channels according to edge orientation. The distance is computed separately for each channel, and the sum of these is the total cost.

Both chamfer and Hausdorff matching can be viewed as special cases of linear classifiers [7]. Let $\mathbf{B}$ be the feature map of an image region, for example a binary edge image. The template $\mathbf{A}$ is of the same size and can be thought of as a prototype shape. The problem of recognition is to decide whether or not $\mathbf{B}$ is an instance of $\mathbf{A}$. By writing the entries of matrices $\mathbf{A}$ and $\mathbf{B}$ into vectors $\mathbf{a}$ and $\mathbf{b}$, respectively, this problem can be written as



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

Figure 2: **Examples of training and test images.** *The following are example image regions used for training and testing a linear classifier:* **(a)** *positive training examples,* **(b)** *test images,* **(c)** *negative training examples containing a hand in different poses,* **(d)** *negative examples containing office scenes as background.*

a linear classification problem with a discriminant function $\langle \mathbf{a}, \mathbf{b} \rangle = c$, with constant $c \in I\!R$. This generalization also permits negative coefficients of $\mathbf{a}$, potentially increasing the cost of cluttered image areas, and different weights may be given to different parts of the shape. Felzenszwalb [7] has shown that a single template $\mathbf{A}$ and a dilated edge map $\mathbf{B}$ is sufficient to detect a variety of shapes of a walking person. A classifier is thus defined by the entries in the matrix $\mathbf{A}$ and in this paper the following classifiers are evaluated (illustrated in figure 3). For each type two sets of templates are generated, one with and one without orientation information. For oriented edges the angle space is subdivided into six discrete intervals, resulting in a template for each orientation channel.

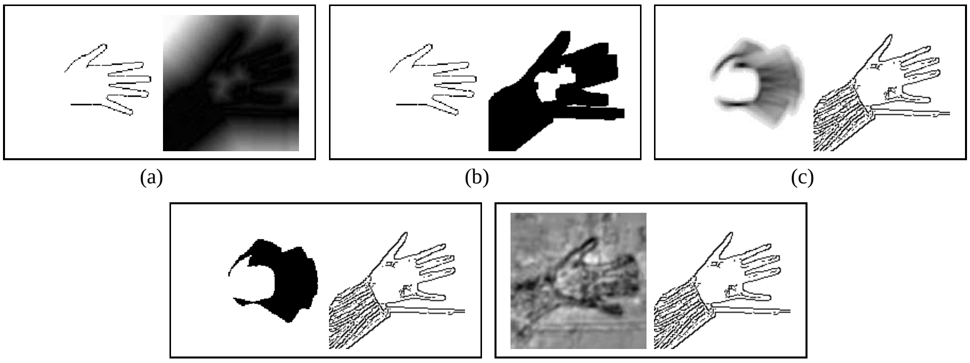
1. Centre template: This classifier uses a single shape template $\mathbf{A}$, generated using the centre of a region in parameter space. Two possibilities for the feature matrix $\mathbf{B}$ are compared. One is the distance transformed edge image in order to compute the truncated chamfer distance [8]. For comparison, the Hausdorff fraction is computed using the dilated edge image [11]. The parameters for both methods are set by testing the classification performance on a test set of 5000 images. Values for the chamfer threshold τ from 2 to 120 were tested, and $\tau = 50$ was chosen, but little variation was observed for values larger than 20. For the dilation parameter δ values from 1 to 11 were compared, and $\delta = 3$ showed the best performance.

2. Marginalized template: In order to construct a classifier which is sensitive to a particular region in parameter space, the template $\mathbf{A}$ is constructed by densely sampling the values in this region, and simultaneously setting the model parameters to these values. The resulting model projections are then pixel-wise added and smoothed. Different versions of matrices $\mathbf{A}$ are compared: (a) the pixel-wise average of model projections, (b) the pixel-wise average, additionally setting the background weights uniformly to a negative value such that the sum of coefficients is zero, and (c) the union of all projections, resulting in a binary template.

3. Linear classifier learnt from image data: The template $\mathbf{A}$ is obtained by learning a classifier as described by Felzenszwalb [7]. A labelled training set containing 1000 positive examples and 4000 negative examples of which 1000 contain the hand in a different pose and 3000 images contain background regions (see figure 2) is used to train a linear classifier by minimizing the perceptron cost function [6].

3.2 Colour features

Given an input image region, define the feature matrix $\mathbf{B}^s$ as the log-likelihood map of skin colour and $\mathbf{B}^{bg}$ as the log-likelihood map of background colour. Skin colour is represented as a Gaussian in (r, g)-space, and the background distribution is modelled as a uniform distribution. A silhouette template $\mathbf{A}$ is defined as containing +1 at locations within the hand silhouette and zero otherwise.



(a) (b) (c)

(d) (e)

Figure 3: **Templates and feature maps used for classification.** *This figure shows the different choices of classifiers* $\mathbf{A}$ *(left) and the corresponding feature maps* $\mathbf{B}$ *(right) used in the experiments.* **(a)** *Centre template with DT of edge image (chamfer)* **(b)** *Centre template with dilated edge image (Hausdorff)* **(c)** *Averaged template with edge image* **(d)** *Union template with edge image* **(e)** *Template learnt from data with edge image.*

Writing these matrices as vectors, a cost function, which corresponds to the log-likelihood [22] can be written as:

$$\mathbf{a}^{\mathrm{T}}(\mathbf{b}^{s} - \mathbf{b}^{bg}) + \mathbf{1}^{\mathrm{T}}\mathbf{b}^{bg} \quad , \tag{1}$$

where the entries in $\mathbf{b}^{bg}$ are constant when using a uniform background model. Thus, correlating the matrix $\mathbf{B} = \mathbf{B}^{s} - \mathbf{B}^{bg}$ with a silhouette template $\mathbf{A}$ corresponds to evaluating the log-likelihood of an input region up to an additive constant. Note that when the distributions are fixed, the log-likelihood values for each colour vector can be pre-computed and stored in a look-up table beforehand. The corresponding templates $\mathbf{A}$ are shown in figure 3 (b).

4 Classifier comparison

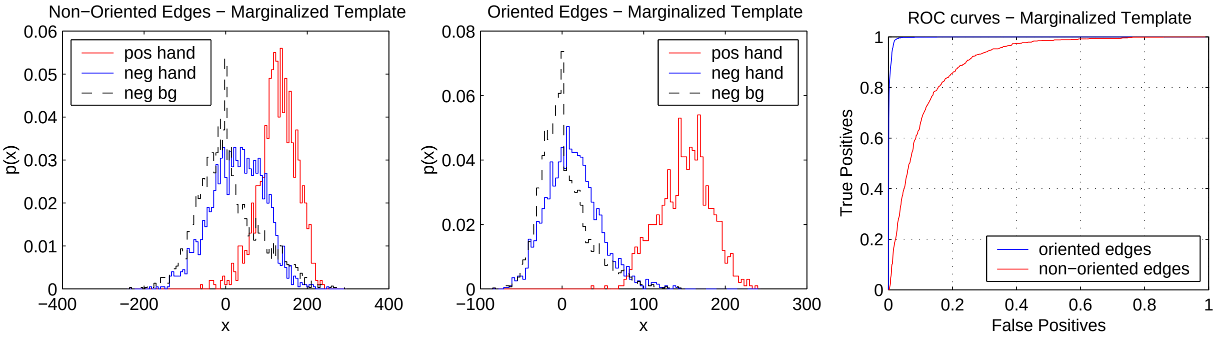
In order to compare the performance of different classifiers, a labelled set of hand images was collected. Positive examples are defined as the hand being within a region in parameter space. For the following experiments this region is chosen to be a rotation of 30 degrees parallel to the image plane. Negative examples are background images as well as images of hands in configurations outside of this parameter region. The evaluation of classifiers is done in three independent experiments for different hand poses, an open hand, a pointing hand and a closed hand. The test data sets each contain 5000 images, of which 1000 are true positive examples. The classifiers are defined by the entries in the matrix $\mathbf{A}$, described in the previous section, and illustrated in figure 3 (a) and (b).

4.1 Edge templates

The following observations were made consistently in the experiments:

- In all cases the use of edge orientation resulted in better classification performance. Including the gradient direction is particularly useful when discriminating between positive examples and negative examples of hand images. This is illustrated in figure 4 (a) and (b), which show the class distributions for non-oriented edges and oriented edges in the case of marginalized templates with non-negative weights. The corresponding ROC curves are shown in 4 (c), demonstrating the benefit of using oriented edges. This shift in the ROC curve is observed in different amounts for all classifiers and is shown in figure 5 (a) and (b).

- In all experiments the best classification results were obtained by the classifier trained on real data. The ROC curves for a particular hand pose (open hand parallel to image plane) are shown in figure 5. At



(a) (b) (c)

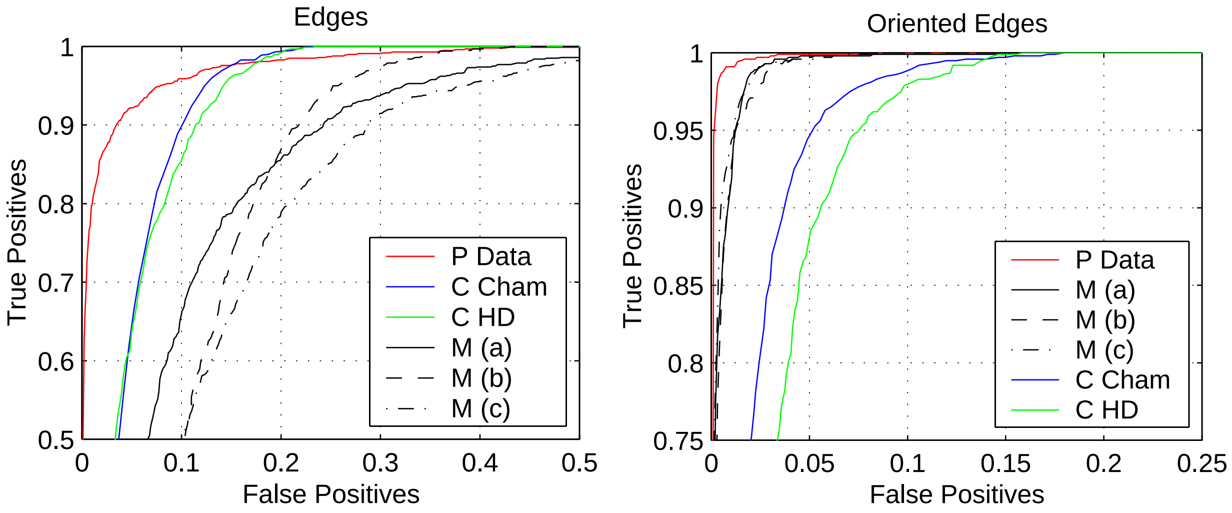
Figure 4: **Including edge orientation improves classification performance.** *This example shows the classification results on a test set using a marginalized template.* **(a)** *histogram of classifier output using edges without orientation information: hand in correct pose (red/light), hand in incorrect pose (blue) and background regions (black,dashed line).* **(b)** *histogram of classifier output using edges with orientation information. The classes are clearly better separated,* **(c)** *the corresponding ROC curve.*

a detection rate of 0.99 the false positive rate was below 0.05 in all experiments.

- Marginalized templates showed good results, also yielding low false positive rates at high detection rates. Templates using pixel-wise averaging and negative weights for background edges were found to perform best when comparing the three versions of marginalized templates. For this template the false positive rates were below 0.11 at a detection rate of 0.99.

- Using the centre template with chamfer or Hausdorff matching showed slightly lower classification performance than the other methods, but in all cases the false positive rate was still below 0.21 for detection rates of 0.99. Chamfer matching gave better results than Hausdorff matching, as can be seen in the ROC curve in figure 5 (b). It should be noted that this result is in contrast to the observations made by Huttenlocher in [10], where Hausdorff matching is shown to outperform chamfer matching in a Monte-Carlo simulation study. However, this may be explained by differences in the implementation details. Whereas the L_1 norm and a threshold value of $\tau = 2$ is used in [10], here the L_2 norm and a threshold value of $\tau = 50$ is used. Values of τ in the range of 2–10 were tested in initial experiments and were found to yield worse results.

The execution times for different choices of templates $\mathbf{A}$ were compared in order to assess the computational efficiency. Computing the scalar product of two vectors of size 128×128 is relatively expensive. However, the computational time can be reduced by avoiding the multiplication of zero valued entries in the matrix $\mathbf{A}$. For chamfer and Hausdorff matching, the template only contains the points of a single model projection. The number of points in the marginalized template depends on the size of the parameter space region it represents. In the experiments it contained approximately 14 times as many non-zero points as a single model template. When using a binary template the dot product computation simplifies to additions of coefficients. If both vectors are in binary form, a further speed-up can be achieved by using binary AND operations. The execution times for correlating 10 000 templates are shown in table 1. The time for computing a distance transform or dilation, which needs to be only computed once for each frame when chamfer or Hausdorff matching is used, is less than 2 ms and is therefore negligible when matching a large number of templates (2.4 GHz Pentium IV). There is a trade-off between computation time and classification performance for the classifiers. When used in a cascaded structure, the detection rate of a classifier needs to be very high, so as not to miss any true positives. In this experiment cham-



(a)

(b)

Figure 5: **ROC curves for classifiers.** *This figure shows the ROC curve for each of the classifiers.* **(a)** *edge features alone, and* **(b)** *oriented edges. Note the difference in scale of the axes. The classifier trained on real image data performs best, the marginalized templates all show similar results, and chamfer matching is slightly better than Hausdorff matching in this experiment. When used within a cascade structure, the performance at high detection rates is important.*

fer and Hausdorff matching, while having a larger false positive rate, are about 10-14 times faster to evaluate than marginalized templates and about 40 times faster than the trained classifier.

4.2 Silhouette templates

The same test data set as for edges was used, and the following observations were made:

- For the test set colour information helps to discriminate between positive examples of hands and background regions. However, there is significant overlap between the positive and negative class examples which contain a hand. Oriented edges are better features to discriminate between the hand in different poses, whereas colour features are slightly better at discriminating between the positive class and background regions.

- Both, centre template and marginalized template show better classification performance than the trained classifier, in particular in the high detection range. At detection rates of 0.99 the false positive rate for the centre template is 0.24, wheras it is 0.64 for the trained classifier. However, the trained classifier shows better performance at separating positive examples from negative example images containing hands. At a detection rate of 0.99, the false positive rate is 0.41 compared to 0.56 for the other two classifiers.

The evaluation can be performed efficiently by precomputing a sum table, $\mathbf{B}^{sum}$, which contains the cumulative sums of costs along the x-direction:

$$\mathbf{B}^{sum}(x,y) = \sum_{i=1}^{x} \left(\log p^{s}(I(i,y)) - \log p^{bg}(I(i,y))\right) \quad , \tag{2}$$

where in this equation the image I is indexed by its x and y-coordinates. p^s and p^{bg} are the skin colour and background colour distributions, respectively. This array only needs to be computed once, and is then used to compute sums over areas by adding and subtracting values at points on the silhouette contour, see figure 6.

Classification Method	Number of points	Execution time	fp at $tp = 0.99$
Chamfer	400	13 ms	0.10
Hausdorff	400	13 ms	0.12
Marginalized Template	5 800	186 ms	0.02
Binary Marginalized Template	5 800	136 ms	0.02
Trained Classifier Template	16 384	524 ms	0.01

Table 1: **Computation times for correlating templates.** *The execution times for computing the dot product of 10 000 image patches of size* 128×128*, where only the non-zero coefficients are correlated for efficiency, measured on a 2.4 GHz Pentium IV machine. The last column shows the false positive rates for each classifier at a fixed detection rate of 0.99.*

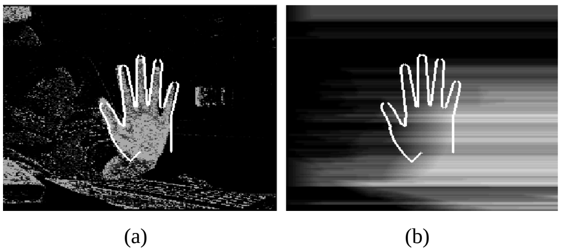

(a) (b)

Figure 6: **Efficient evaluation of colour likelihoods. (a)** *The skin colour log-likelihood image encoded as a greyscale image. Higher intensity corresponds to higher likelihood of skin vs. non-skin colour.* **(b)** *The sum table contains the cumulative sum of values in image (a) along the* x*-direction. The sum of values in within an area can be efficiently computed by adding and subtracting values at silhouette points only. The greyscale intensities are scaled to the range [0,255] in both images.*

It is a convenient way to convert area integrals into contour integrals, and is related to the summed area tables of Crow [5], the integral image of Viola and Jones [27] or the integration method based on Green's theorem of Jermyn and Ishikawa [14]. Compared to integral images the sum over non-rectangular regions can be computed more efficiently, and in contrast to the technique of Jermyn and Ishikawa the contour normals are not needed. In contrast to these methods, however, the model points need to be connected pixels, e.g. obtained by line-scanning a silhouette image. The computation time for evaluating 10 000 templates is reduced from 524 ms to 13 ms for a silhouette template of 400 points. As the computation of the sum table $\mathbf{B}^{sum}$ is only computed once, this is negligible when matching a large number of templates.

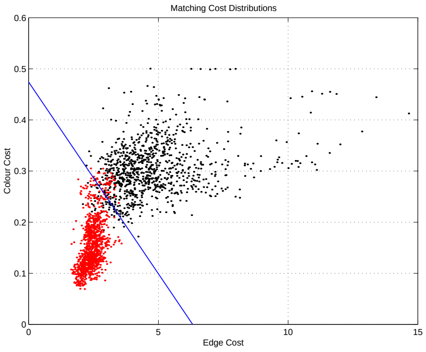


Figure 7: **Determining the weighting factor in the cost function.** *The distribution of edge and colour cost values for a number of positive (lower left) and negative training examples (upper right) is shown, and the linear classifier found using a maximum margin linear classifier. The weighting factor is set to the negative inverse of the slope of this line.*

4.3 Combining edge and colour information

In the previous sections similarity measures based on edge and colour information have been derived. When combining the two features, they can be stacked into a single 2D observation vector $\mathbf{z} = (\mathbf{z}^{edge}, \mathbf{z}^{col})^{\mathrm{T}}$. In a first approach, the edge and colour cost terms are computed for a number of test images. Figure 7 shows a graph of the distributions of the cost vectors for 1000 positive and 1000 negative example image subregions. Positive examples correspond to a hand being in a pose corresponding to the parameter range represented by the classifier. Negative examples correspond to the hand in different poses and background regions. The plot shows that for the test images that the class overlap is not large and a linear discriminant is used to separate the two classes, yielding a total misclassification rate of 4.8% on this data set. Using a linear discriminant corresponds to simply using a weighted sum of the edge and colour cost terms [4], where the weighting factor is derived from the data directly in order to yield optimal classification performance on a test set. Experiments for three different classifiers, corresponding to the hand in different poses, but at the same scale, show that this weighting factor varies little for different templates.

5 Experimental results

This section demonstrates the use of the template based classifiers in detection and pose estimation tasks. In these experiments the chamfer distance cost function is used to compute the edge cost.

5.1 Detection of a single hand pose

In order to test the integration of shape and colour information a set of 500 templates was generated, corresponding to 100 discrete orientations and five different scales. The templates are matched to an image by translating it over the image at a 6-pixel resolution in x and y-direction. The weighted cost function was used to detect a hand in the following scenarios:

- A hand in front of cluttered background, which contains little skin colour: In this case the hand is difficult to detect using edge information alone. The colour likelihood, however allows for correct detection (See top row of figure 8).
- A hand in front of a face: In this example there is not enough skin colour information to detect the hand, however the hand edges are still visible and are used as features to correctly locate the hand (See bottom row of figure 8).

This illustrative example shows that using a robust cost function can improve the performance of hand detection or tracking algorithms that use intensity or skin colour edges alone.

In the second experiment, an input sequence of 640 frames was recorded. The pose is an open hand, parallel to the image plane, and it moved with 4 DOF; translation in x, y, and z-direction, as well as rotation around the z-axis. The task is made challenging by introducing a cluttered background with skin-coloured objects. The hand motion is fast, and during the sequence the hand is partially and fully occluded, as well as out of the camera view. The same set of 500 templates is used to search for the best match over the image.

Figure 9 shows typical results for a number of frames of this sequence and figure 10 shows the position error measured against manually labelled ground truth. The RMS error over the complete sequence for the frames in which the hand was detected, was 3.7 pixels. The main reason for the magnitude of this error is the coarse search at 6-pixel resolution in translation space. Assuming a uniform distribution on the hand location in the image, the expected RMS error is larger than 1.5 pixels. For comparison, a tracker based using the *unscented Kalman filter* [16, 23] was also run on this input sequence, but it was not able to track the hand for more than 20 frames. One of the reasons for the loss of track is that the motion in this sequence is fast and abrupt, so that neither a constant velocity nore a constant acceleration model was accurate enough.

5.2 Hierarchical detection

The tree-based detection method was tested on real image data. This corresponds to the initialization stage in the hierarchical filter [22, 24]. Figure 11 illustrates the operation of the classifiers at different levels of the tree. In this case the classifiers are based on oriented edges using the chamfer distance and skin colour silhouette. The classifiers at the upper levels correspond to larger regions of parameter space, and are thus less discriminative. The figure shows examples of accepted and rejected templates at different tree levels for a different input image. It can be seen that as the search proceeds, the difference between accepted and rejected templates decreases, and the quality of the best matches increases. The tree contains 8748 different templates, corresponding to a pointing hand, restricted to rigid motion in a hemisphere. The tree consists of four levels, corresponding to a search over 972 discrete angles and nine scales, and a search over translation space

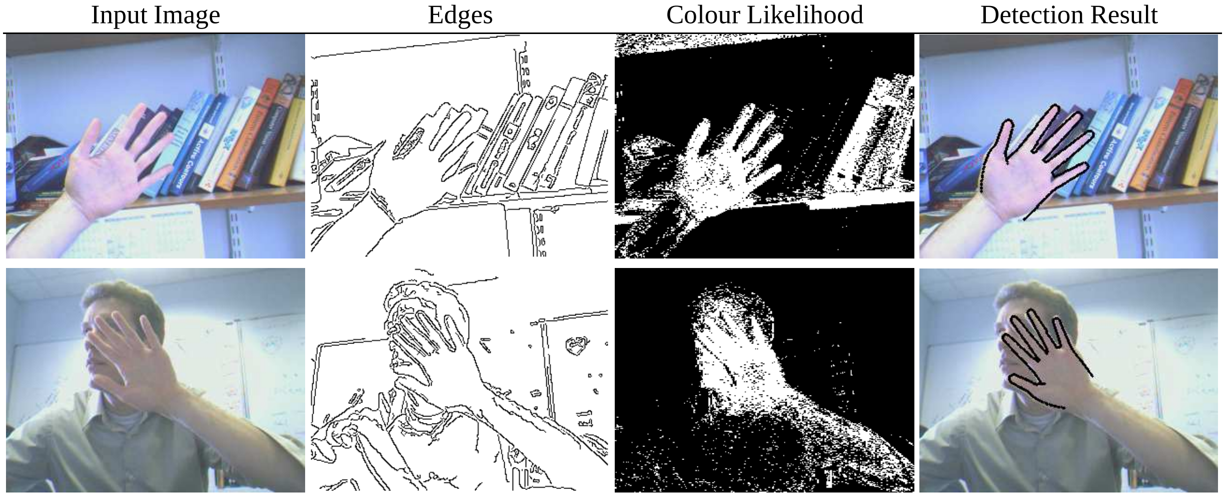


Figure 8: **Detection with integrated edge and colour features.** *This illustrative example shows how a cost function that uses edge and colour features improves detection. For each input image the best match is shown in the last column.* **(Top row)** *Hand in front of cluttered background, and* **(bottom row)** *hand in front of face.*

at single pixel resolution.

6 Conclusion

In this paper the concept of a hierarchical cascade of classifiers for locating articulated objects was introduced and a number of different similarity measures for template matching with shape variation were considered. If a training set of shapes is available, a linear classifier can be trained which has high discriminative power. It can give different weights to different parts of the shape and is able to penalize background edges. However, it is also expensive to evaluate. Marginalized templates have been introduced as a method to use an object model to construct a classifier, which is specifically "tuned" to a specific region of parameter space. They can be efficiently evaluated, for example by approximating them with a binary valued template. Finally, templates generated from a single contour ("centre templates" in the experiments), can be used to classify shapes. However it is necessary to preprocess the edge image first, e.g. by a dilation operation or a distance transform, in order to be tolerant to shape variation. The main advantage of this approach is that it is very efficient because the pre-processing step is only required once per image and matching a single template is fast.

The motivation of this research has been work on hand tracking, in which we seek to combine the merits of efficient detection and tracking. Motivated by the success of tree-based detection, the parameter space is discretized to generate a tree of templates which can be used as classifiers. Even though their performance has been shown to be not as good as classifiers learnt from image data, they have the advantage of being easy to generate and being labelled with a known 3D pose, permitting their use in a model-based tracking framework.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 432–439, Madison, WI, June 2003.

[2] S. Baker and S. K. Nayar. Pattern rejection. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 544–549, San Francisco, CA, June 1996.

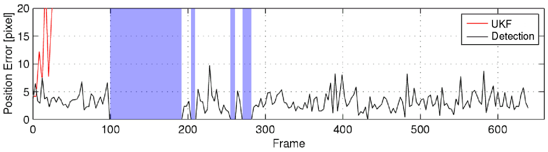


Figure 10: **Error comparison between UKF tracking and detection.** *This figure shows the error performance of the UKF tracker and detection on the image sequence in figure 9. The hand position error was measured against manually labelled ground truth. The shaded areas (blue) indicate intervals in which the hand is either fully occluded or out of camera view. The detection algortihm successfully finds the hand in the whole sequence, whereas the UKF tracker using skin-colour edges is only able to track the hand for a few frames. The reasons for the loss of track is that the hand motion is fast between two frames and that skin-colour edges cannot be reliably found in this input sequence.*

[3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977.

[4] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 232–237, Santa Barbara, CA, June 1998.

[5] F. C. Crow. Summed-area tables for texture mapping. In *Proc. SIGGRAPH*, number 3, pages 207–212, July 1984.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, second edition, 2001.

[7] P. F. Felzenszwalb. Learning models for object recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 56–62, Kauai, HI, December 2001.

[8] D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. 6th European Conf. on Computer Vision*, volume II, pages 37–49, Dublin, Ireland, June/July 2000.

[9] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to non-linear and non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140:107–113, 1993.

[10] D. P. Huttenlocher. Monte Carlo comparison of distance transform based matching measure. In *ARPA Image Understanding Workshop*, pages 1179–1183, New Orleans, LA, May 1997.

[11] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intell.*, 15(9):850–863, 1993.

[12] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, pages 93–101, Berlin, May 1993.

[13] M. Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. on Computer Vision*, pages 343–356, Cambridge, UK, April 1996.

[14] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries. In *Proc. 7th Int. Conf. on Computer Vision*, volume I, pages 20–27, Corfu, Greece, September 1999.

[15] N. Jojic, N. Petrovic, B. J. Frey, and T. S. Huang. Transformed hidden markov models: Estimating mixture models and inferring spatial transformations in video sequences. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 26–33, Hilton Head, SC, June 2000.

[16] S. J. Julier. *Process Models for the Navigation of High-Speed Land Vehicles*. PhD thesis, Department of Engineering Science, University of Oxford, 1997.

[17] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. 6th European Conf. on Computer Vision*, volume 2, pages 3–19, Dublin, Ireland, June 2000.

[18] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *Transactions on Image Processing*, 6(1):103–113, January 1997.

[19] S. Romdhani, P. H. S. Torr, B. Schölkopf, and A. Blake. Computationally efficient face detection. In *Proc. 8th Int. Conf. on Computer Vision*, volume II, pages 695–700, Vancouver, Canada, July 2001.

[20] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 746–751, Hilton Head, SC, June 2000.

[21] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. 9th Int. Conf. on Computer Vision*, volume II, pages 750–757, Nice, France, October 2003.

[22] B. Stenger. *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. PhD thesis, University of Cambridge, Cambridge, UK, 2004.

[23] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 310–315, Kauai, HI, December 2001.

[24] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proc. 9th Int. Conf. on Computer Vision*, volume II, pages 1063–1070, Nice, France, October 2003.

[25] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Learning a kinematic prior for tree-based filtering. In *Proc. British Machine Vision Conference*, volume 2, pages 589–598, Norwich, UK, September 2003.

[26] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. Journal of Computer Vision*, 48(1):9–19, June 2002.

[27] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 511–518, Kauai, HI, December 2001.

[28] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *Proc. 9th Int. Conf. on Computer Vision*, volume I, pages 353–360, Nice, France, October 2003.

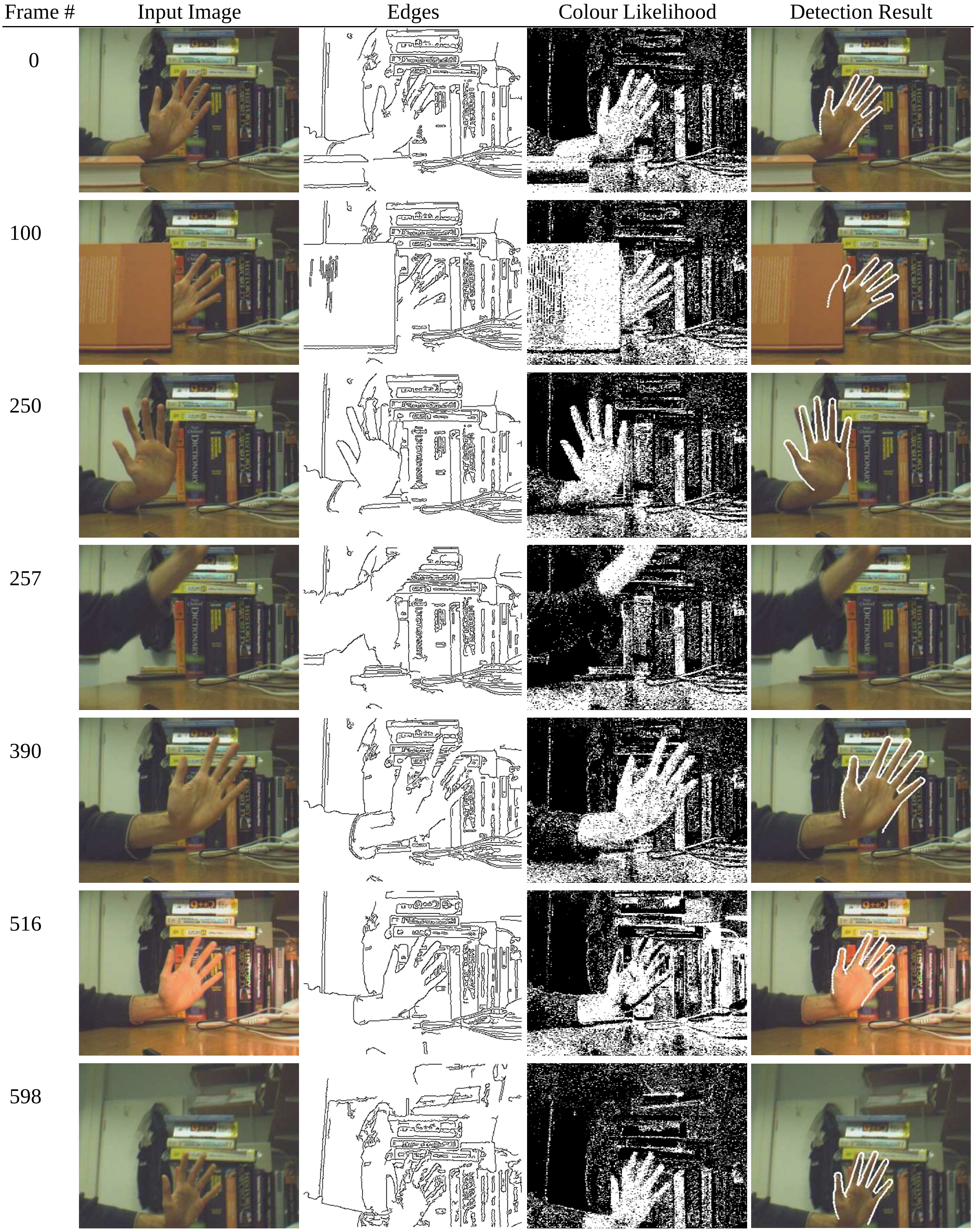


Figure 9: **Detection results using edge and colour information.** *This figure shows successful detection of an open hand moving with 4 DOF. The first two columns show the frame number and the input frame, the next two columns show the Canny edge map and the skin colour likelihood. The last column shows the best match superimposed, if the likelihood function is above a constant threshold. The sequence is challenging because the background contains skin-coloured objects* **(frame 0)** *and motion is fast, leading to motion blur and missed edges. The detection handles some partial occlusion* **(frame 100)**, *recovers from loss of track* **(frames 257, 390)**, *can deal with lighting changes* **(frame 516)** *and unsteady camera motion* **(frame 598)**.

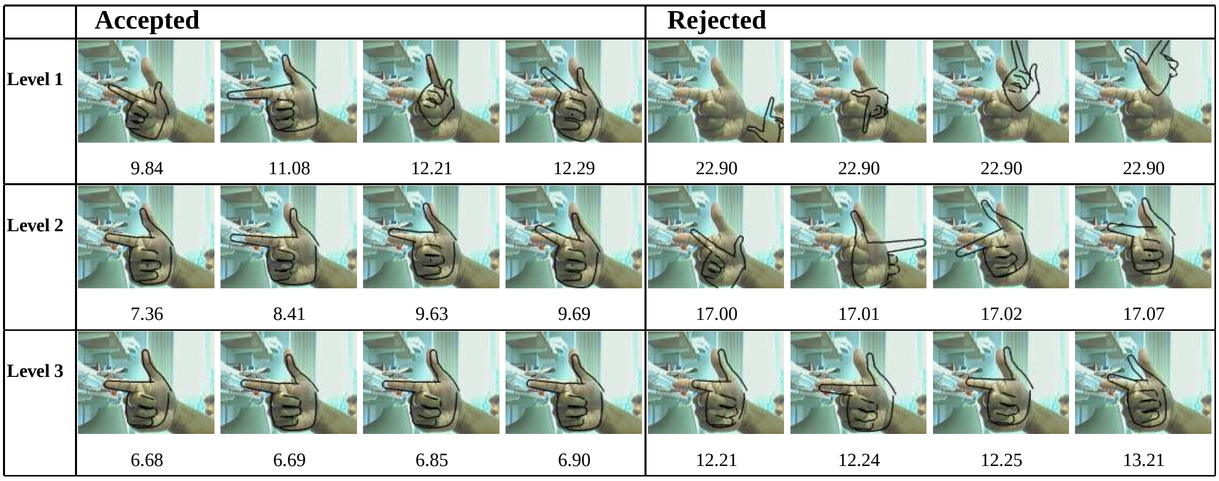


Figure 11: **Search results at different levels of the tree.** *This figure shows typical examples of accepted and rejected templates at levels 1 to 3 of the tree, ranked according to matching cost shown below. As the search is refined at each level, the difference between accepted and rejected templates decreases.*