Using pen-based outlines for object-based annotation and image-based queries

Lambert Schomaker
Edward de Leau
Louis Vuurpijl

NICI, Nijmegen Institute for Cognition and Information
University of Nijmegen, P.O.Box 9104
6500 HE Nijmegen, The Netherlands
Tel: +31 24 3616029 / Fax: +31 24 3616066
schomaker@nici.kun.nl
hwr.nici.kun.nl

Nijmegen Institute for Cognition and Information

Cognitive Engineering Section

group at NICI :

- on-line & off-line HWR approaches
- Multiple Agents in Pattern Recognition (MAPR)
- Information Retrieval/Information Filtering
- hybrid (NN/AI) modeling
- Content-Based Image Retrieval

- image-based retrieval & the user
- design
- pattern recognition
- performance

- **There are already quite a few systems available on WWW, but:**

- **What do users want?**

Question	*Yes*	*No*	**NA**
"Did you need an image ..."			
"...with a particular object on it?"	**122**	**41**	**7**
"...with a particular color on it?"	**25**	**137**	**8**
"...with a particular texture on it?"	**23**	**137**	**10**

(results of a WWW questionnaire, N=170 responses)

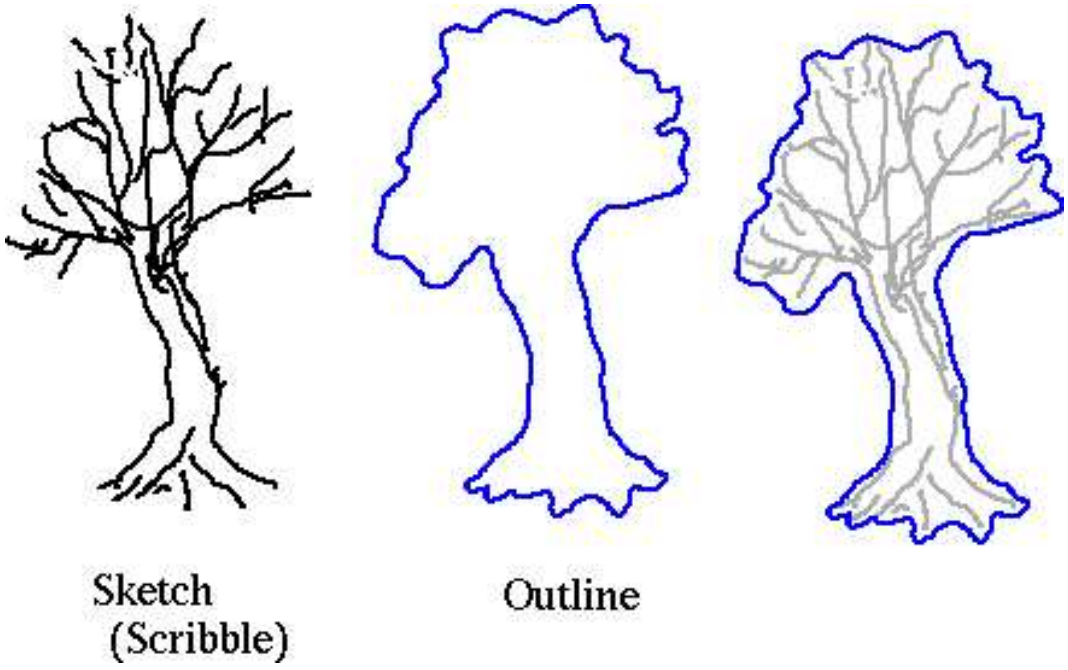
what do users want?

- **Object search!**
  often: the 'basic categories' (Rosch, 1972)
  cf. Hoenkamp, Schomaker & Stegeman, SIGIR'99

- **Not: 'feature configurations'**
  or 'layout structures'

queries and matching methods in image-based search

	Query	*Matched with:*	*Matching algorithm*
A	keywords	manually provided textual image annotations	free text and information-retrieval (IR) methods
B	keywords	textual and contextual information in the image neighbourhood	free text and IR methods
C	exemplar image	image bitmap	template matching or feature-based
D	layout structure	image bitmap	texture and color segmentation
E	object outline	image bitmap, contours	feature-based
F	object sketch	image bitmap	feature-based

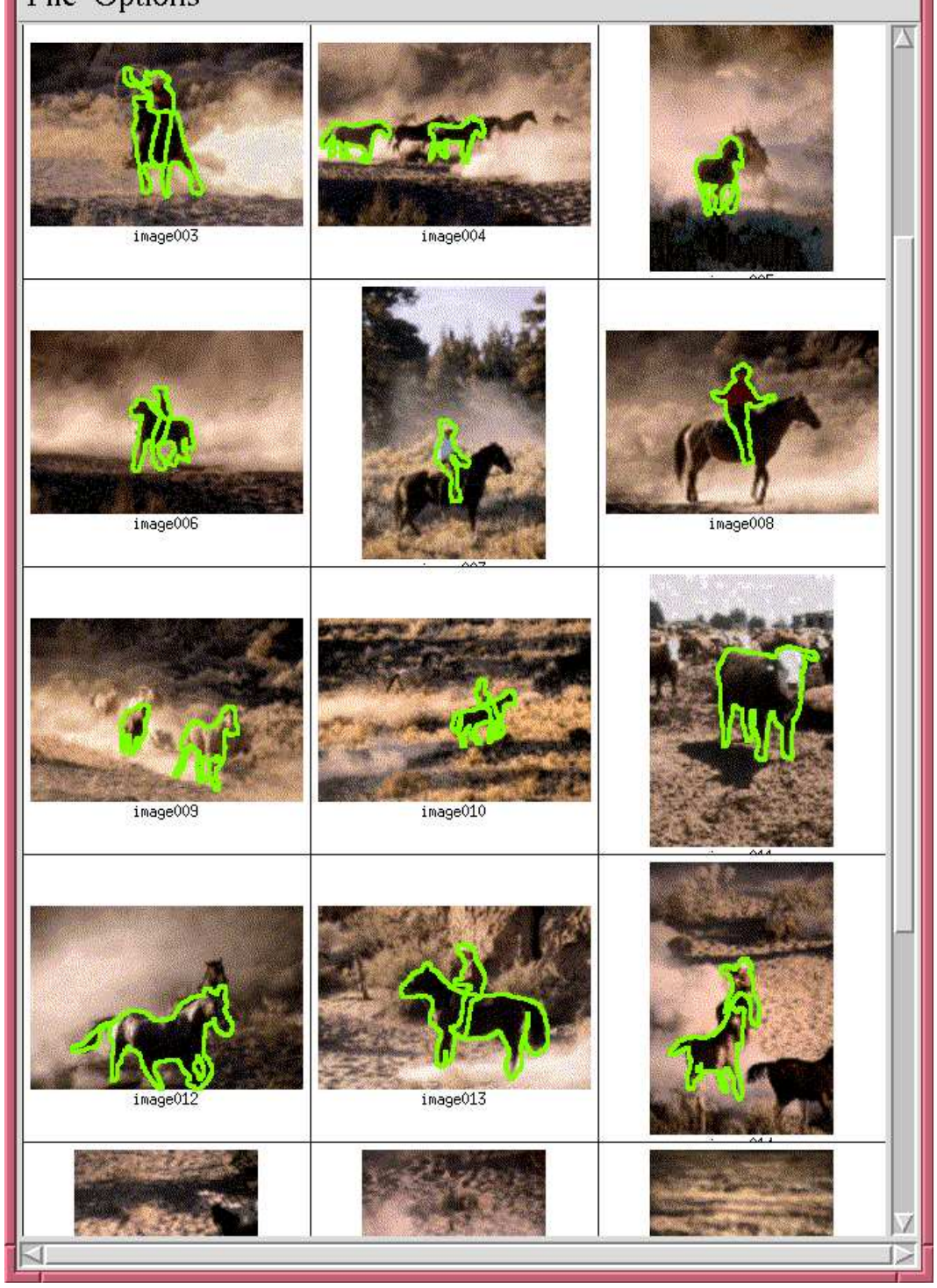


'outline' = closed curve drawn around an object on a photograph

questions:

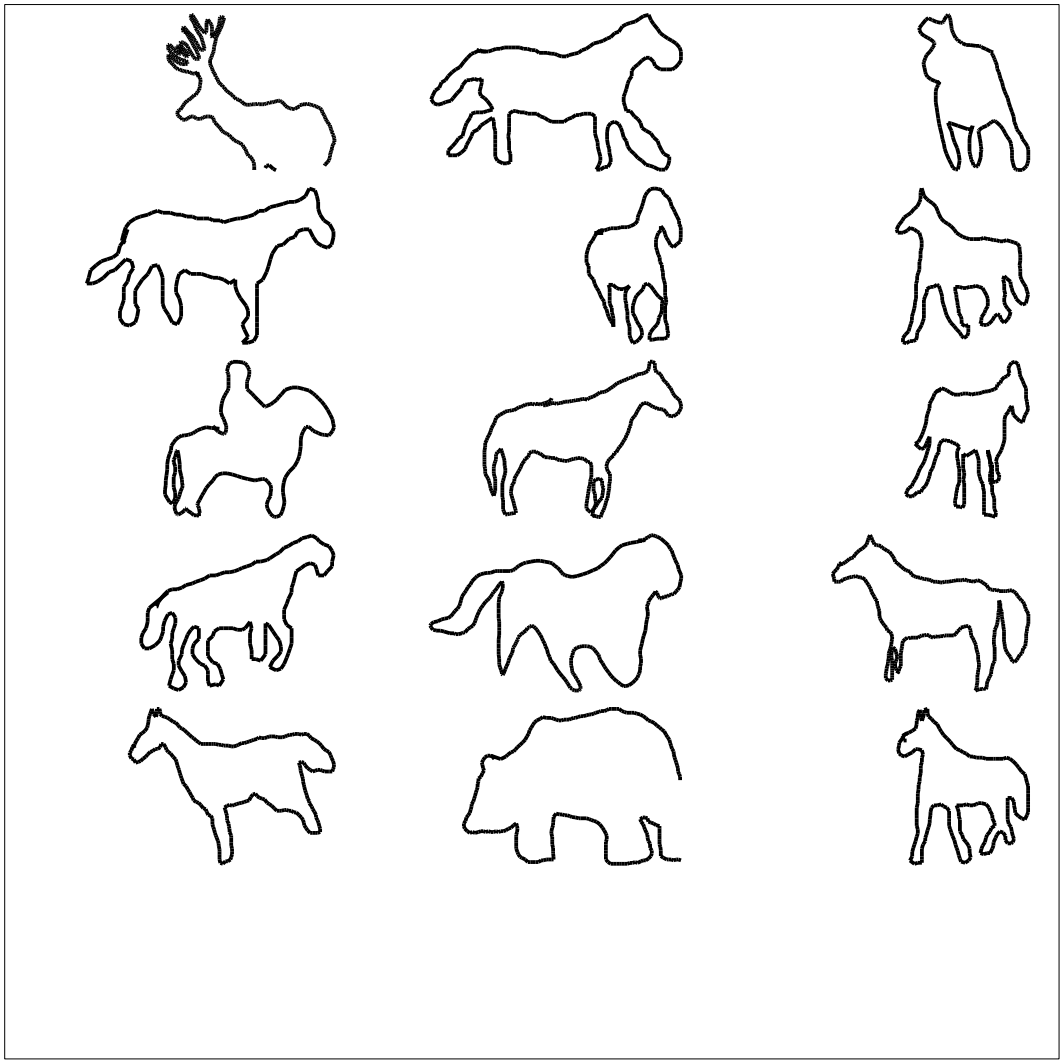
- are the users able to produce the queries?
- do they like to use the query method?
- what classification performance is required? (and how to measure performance?)
- is the system able to explain the results? (Picard: 'explainable features')
- can the system learn from previous queries in a user community?

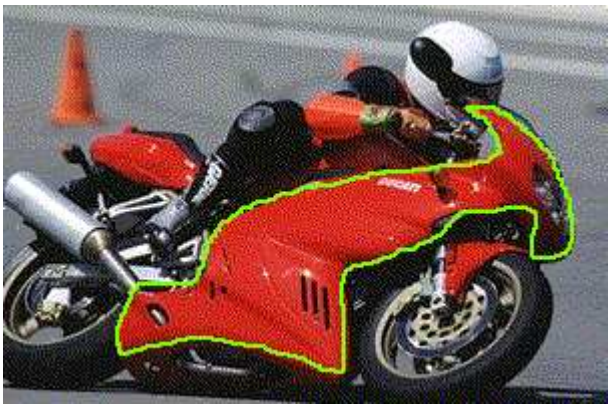
1. focus on object-based representations and queries
2. focus on photographic images with identifiable objects for which a verbal description can be given
3. exploit human perceptual abilities in the user
4. exploit human fine motor control: use a pen to draw object outlines
5. allow for incremental annotation of image material (to obtain PR bootstrap)
6. start with a limited content domain



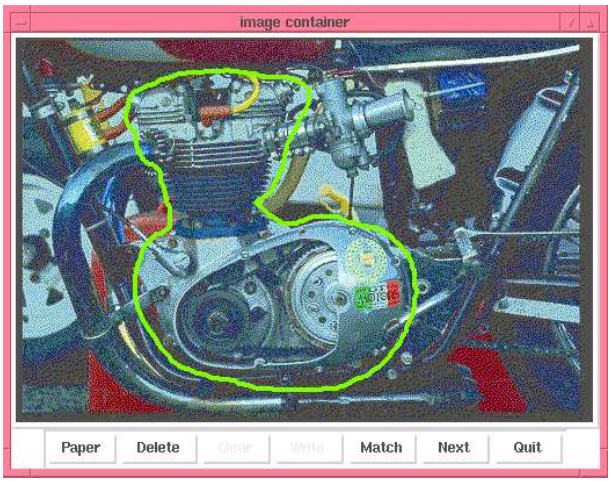
(multiple outlines per photograph are allowed)

outlines

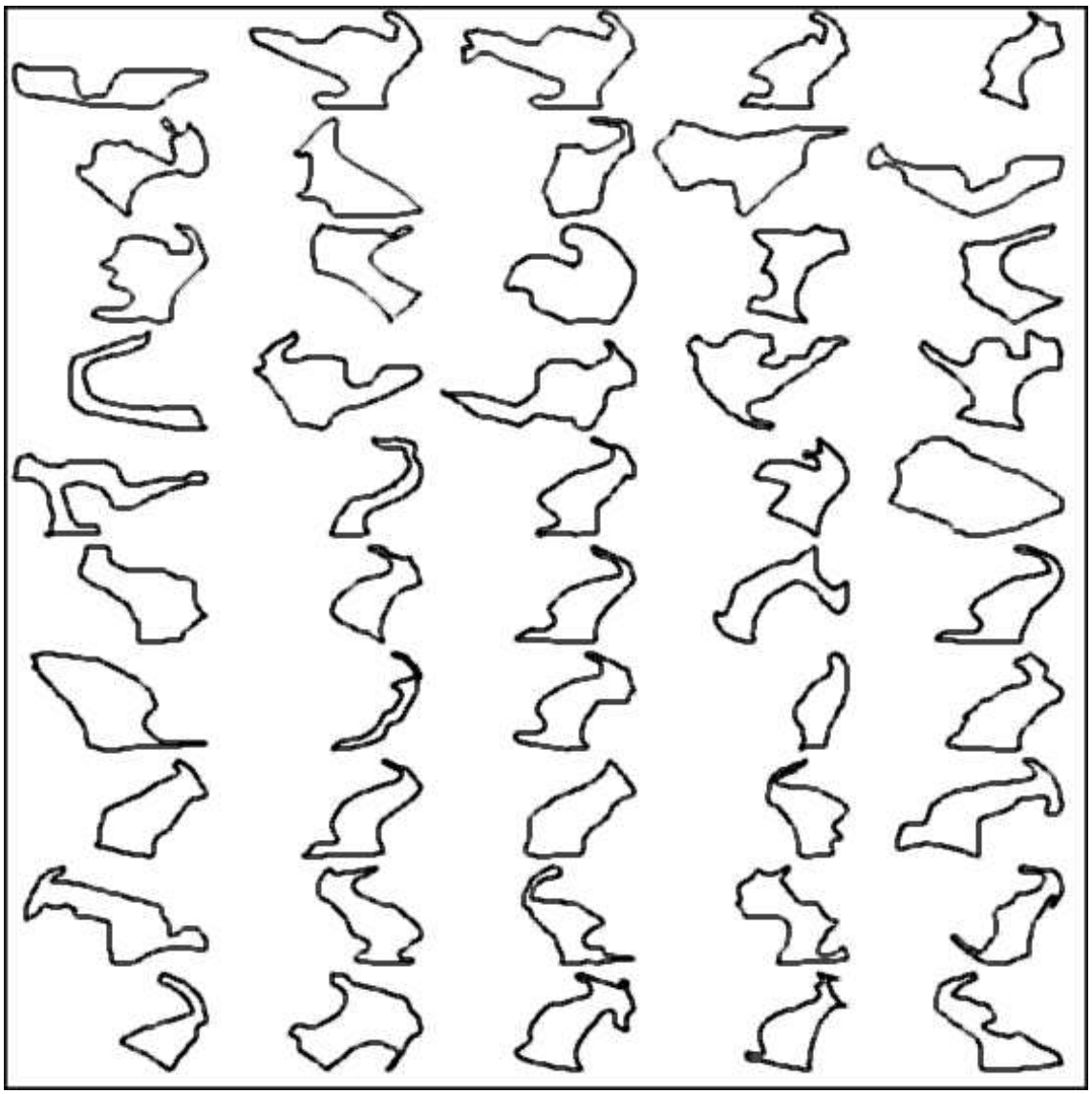




(note the distribution of points of high curvature along the outline)

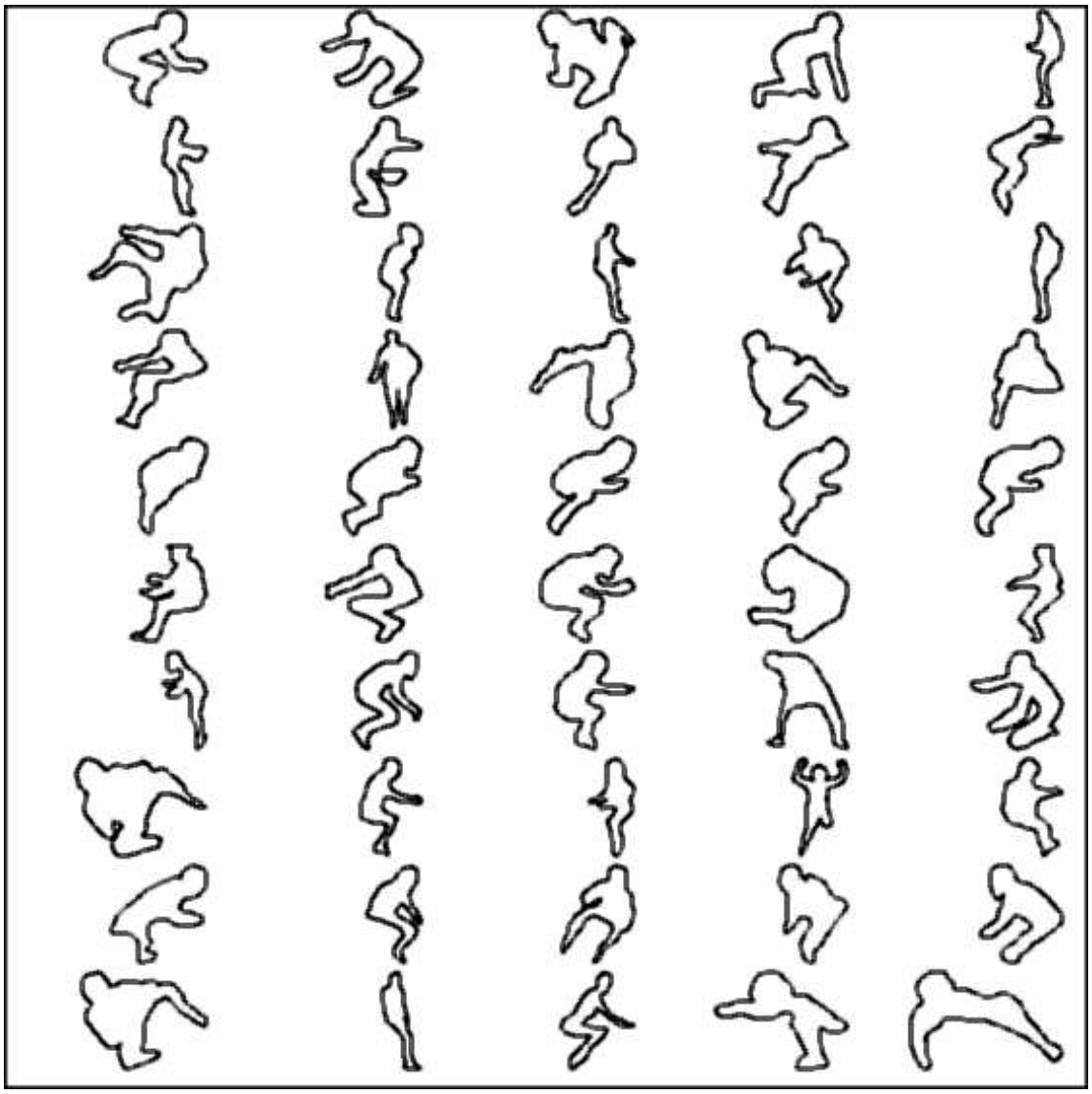


A query to find an *engine*

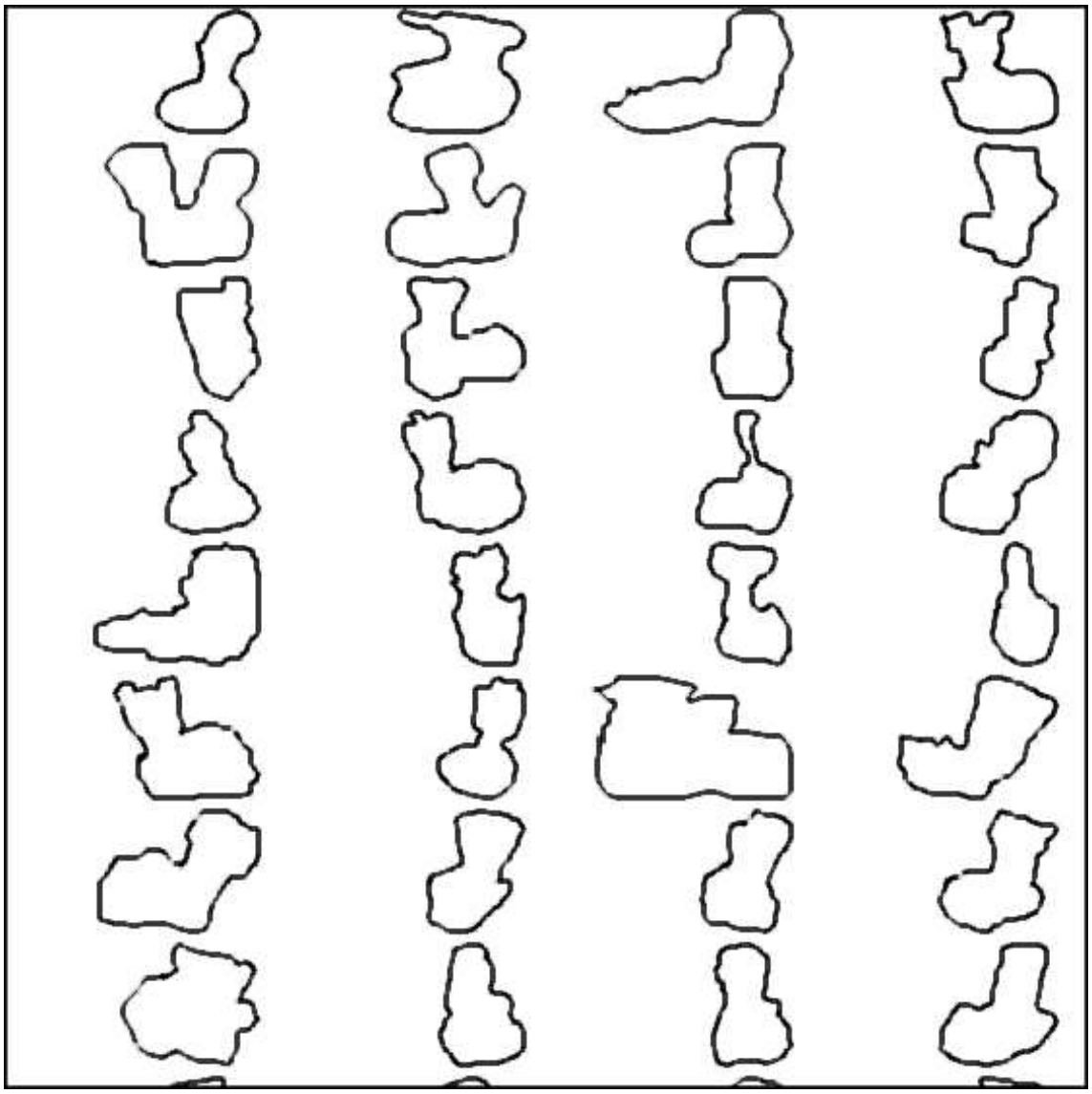


nici
Nijmegen Institute for Cognition and Information
Cognitive Engineering Section

driver shapes

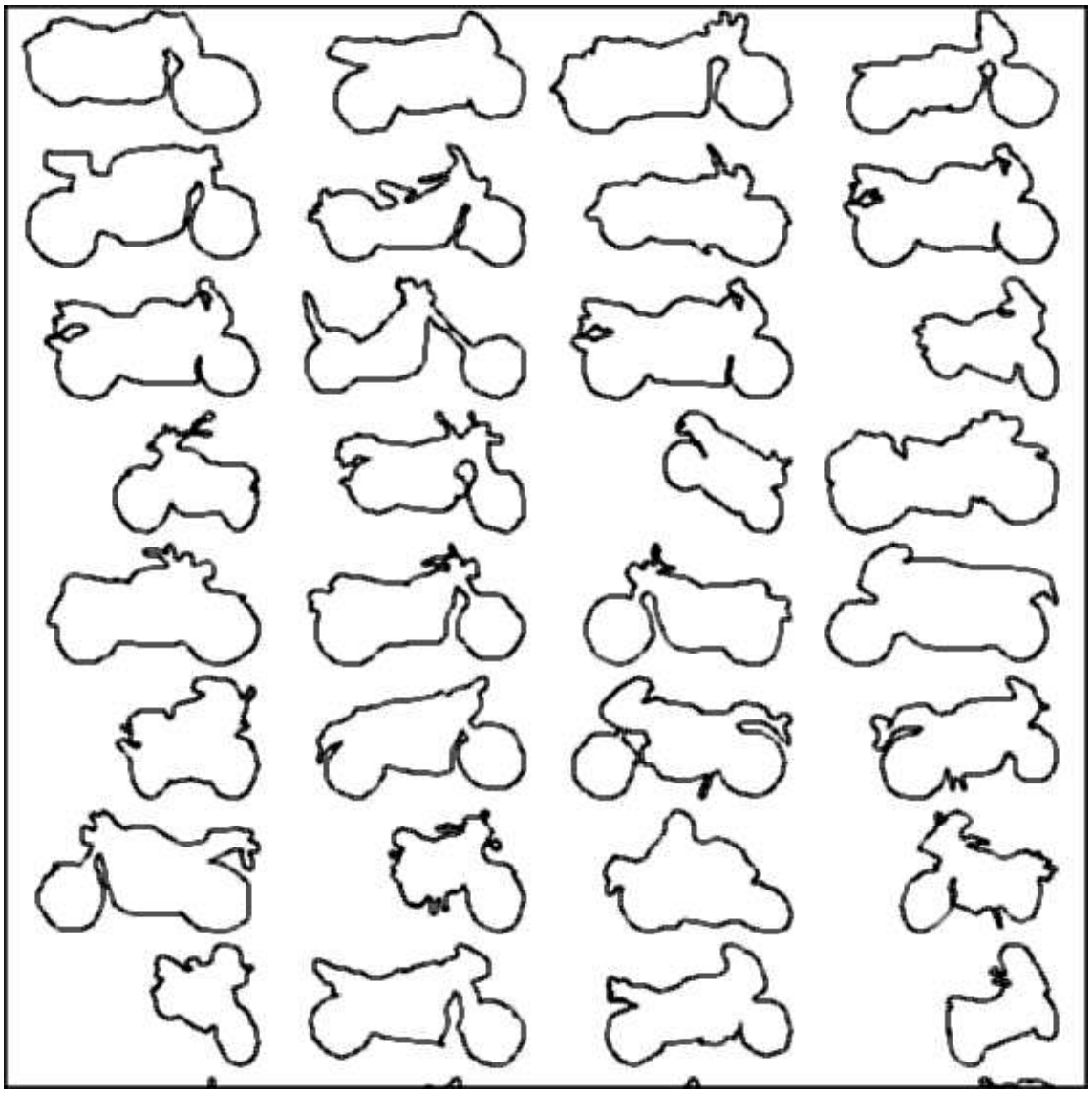


engine shapes



nici
Nijmegen Institute for Cognition and Information
Cognitive Engineering Section

frame shapes



matching possibilities

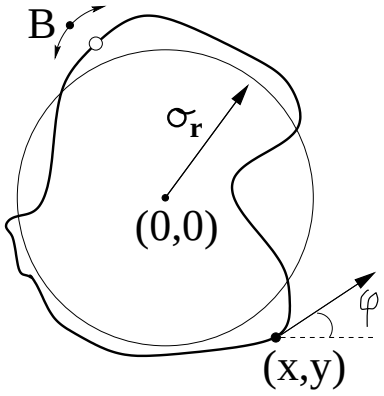
- **(a) match the query outline $(\vec{\mathbf{x}}, \vec{\mathbf{y}})$ with all outlines which are present in the database**

- **(b) match the image $I(x, y)$ content within the outline $(\vec{\mathbf{x}}, \vec{\mathbf{y}})$ with existing templates in the database,**

- **(c) match a query outline with image edges $\Delta I(x, y)$ of unseen photographs (!)**

Simple 1-NN matching will be used for all feature categories.

outline features



The raw outline is resampled to a fixed number of samples (100). The center of gravity is translated to $(0, 0)$, the size is normalized to an rms radius (σ_r) of one, yielding the normalized outline $(\hat{\vec{x}}, \hat{\vec{y}})$. From the starting point B, the matching process will try both clockwise and counter-clockwise directions, retaining the best result of both match variants. Other normalizations such as left/right or up/down mirroring are optional. In addition, the running angles $(cos(\phi), sin(\phi))$ are added as feature group, as well as the histogram $p(\phi)$.

image features

The following 68 features were derived from the pixels within the closed object outline:

color centroids The center of gravity for each of the RGB-channels. This gives 6 features: $R(x,y)$, $G(x,y)$ and $B(x,y)$

color histogram The histogram of the occurrence of 8 main colors: black, blue, green, cyan, red, magenta, yellow and white

intensity histogram A histogram for 10 levels of pixel intensity

RGB statistics The minimum and maximum values of each of the RGB-channels, and their average and standard-deviation (12 features)

texture descriptors A table of five textures was used, with five statistical features each (25 features)

invariant moments Seven statistical high-order moments[**?**] which are invariant to size and rotation

data set

Data set: 200 mixed JPEG and GIF photographs of motor bicycles. Within this set, 750 outlines were drawn around image parts in the following classes: exhaust, wheels, engine, frame, pedal, fuel tank, saddle, driver, mirror, license plate, bodyworks, head light, fuel tank lid, light, rear light, totalling 15 object classes with 50 different outline samples of each object

outline matching & within-outline image matching

Results are represented as the average percentage of correct hits in the top-10 hit list (P_{10}), averaged over $n = 50$ outline instances per class, of which each was used as a probe in nearest-neighbour matching. The query itself was excluded from the matching process.

Query	I. P_{10} (%) $(\hat{x}, \hat{y})$	II. P_{10} (%) $(cos\phi, sin\phi)$	III. P_{10} (%) $p(\phi)$	IV. P_{10} (%) $image-based$
wheels	77.6	81.8	36.0	58.2
exhaust	75.4	79.4	34.0	34.6
engine	57.0	51.4	31.6	49.6
frame	52.0	33.8	38.8	69.4
pedal	47.4	47.2	22.8	33.0
driver	43.6	43.4	20.2	50.2
saddle	41.4	39.2	15.0	20.2
fuel tank	41.4	43.2	23.2	22.8
mirror	40.6	39.8	11.2	22.4
license plate	36.0	47.8	30.2	21.8
bodywork	31.0	26.6	14.4	22.4
head light	30.6	38.2	13.2	30.4
fuel tank lid	29.6	35.8	25.8	23.4
light	21.6	19.4	11.0	27.4
rear light	14.8	14.8	9.0	33.0

outline vs edge matching

Ultimately, one will want to use the set of outlines to perform object classification in unseen images, for which only the 'bottom-up' edge representation can be computed. Assuming that scale and translation are already approximately correct, how well can we match the human-generated outlines with the edges?

For each point i on a raw outline (X_i, Y_i), a convolution is calculated as follows. Let $\Delta I(x, y)$ be an estimate of the absolute and smoothed derivative of the luminance gradient of an image $I(x, y)$, averaged over a number of suitable directions. Then the local match between an outline point (X, Y) and the edge representation of the image can be calculated as:

$$M_{X_i Y_i, \Delta I} = \sum_{\delta_x=-w}^{w} \sum_{\delta_y=-w}^{w} \frac{\Delta I(X_i + \delta_x, Y_i + \delta_y)}{\sqrt{\delta_x^2 + \delta_y^2}} \qquad (1)$$

nici
Nijmegen Institute for Cognition and Information
Cognitive Engineering Section

outline+edge matching

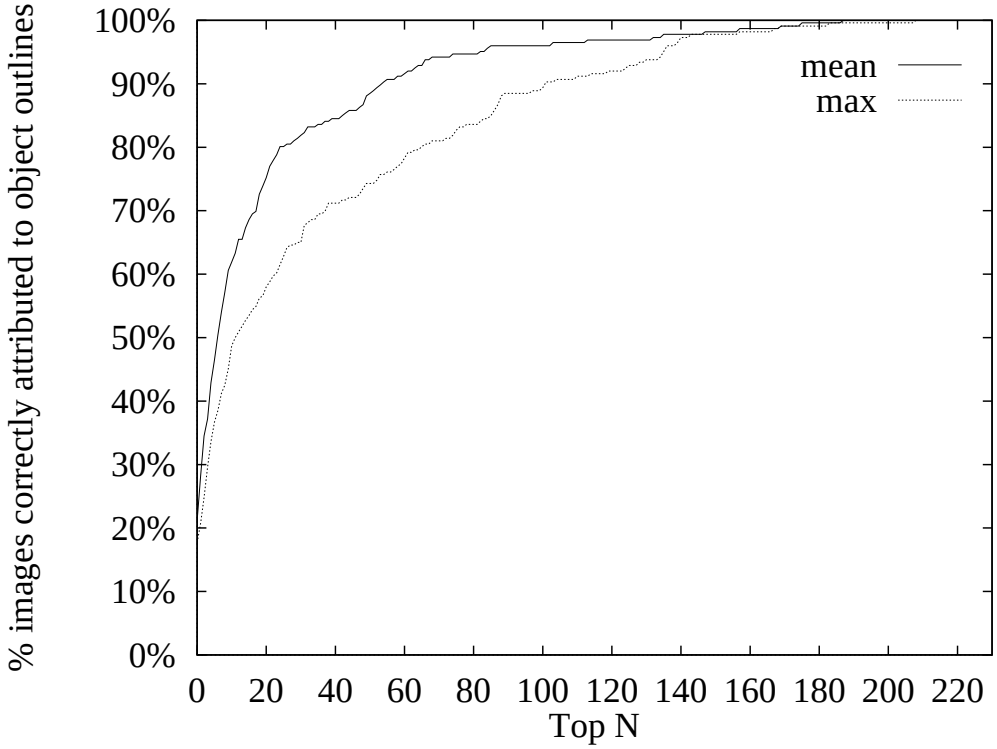


Figure 1: Results for the matching process between human-drawn outlines and bottom-up calculated image edges as a percentage of outline instances which are correctly associated with their original image. The two curves represent the results of sorting the hit list on mean convolution output $\mathcal{M}$ (solid line) or on the maximum value of $\mathcal{M}$ (stippled). This performance measure differs from Table **??** because here instances are matched as opposed to classes.

- promising results (esp. when compared to HWR problems)
- succesfully applied to a set of aircraft images
- computation time...
- refinement of edge preprocessing will improve the 'bottom-up' search for outlines in unseen images
- domain-dependent and object-dependent use of features: ideal environment for the multiple-agent paradigm
- ongoing work: S/N ratios for mouse & pen-based outlines