Johann Wolfgang Goethe-Universität
Frankfurt am Main

Fachbereich Informatik

MASCOT: A Mechanism for Attention-based Scale-invariant Object Recognition in Images

B. Arlt, R. Brause, E. Tratar

{arlt, brause, tratar}@informatik.uni-frankfurt.de

ABSTRACT

The efficient management of large multimedia databases requires the development of new techniques to process, characterize, and search for multimedia objects. Especially in the case of image data, the rapidly growing amount of documents prohibits a manual description of the images' content. Instead, the automated characterization is highly desirable to support annotation and retrieval of digital images. However, this is a very complex and still unsolved task.

To contribute to a solution of this problem, we have developed a mechanism for recognizing objects in images based on the *query by example* paradigm. Therefore, the most salient image features of an example image representing the searched object are extracted to obtain a scale-invariant object model. The use of this model provides an efficient and robust strategy for recognizing objects in images independently of their size. Further applications of the mechanism are classical recognition tasks such as scene decomposition or object tracking in video sequences.

Keywords

Attention-based object recognition, scale-invariant object model, image databases.

1. INTRODUCTION

In the recent years, the distribution and manipulation of multimedia documents have become very important information processing tasks. Especially the rapidly growing amount of images in multimedia databases, digital libraries, internet, newspaper archives, etc. prohibits a manual characterization of the images' content by humans. Instead, new techniques for the automated recognition, retrieval, and annotation of image data are required.

Presently, the search for a specific image in a large database is done by searching the textual annotations related to the images. If the database already contains several thousands of images and/or new images are added frequently, it becomes impossible to completely characterize each image manually. Thus, a mechanism is needed which provides a method to access the image content directly and indexes the images automatically.

In case of searching an image which is showing an object or a scene "unknown" to the system (i.e. no annotation related to the searched image content exists in the database) the indexing method fails. Consequently, we have to apply a different strategy called *query by example*. Here, an example image is presented to the system and compared to the images in the database; the search result consists of those images which are most similar to the example. This technique is used by several commercial (*QBIC* [5], *Virage* [10]) and non-commercial image query systems (*MARS* [8], *VisualSEEk* [9]). However, the applied recognition algorithms have serious problems to detect objects in the presence of occlusion. Furthermore, it is very difficult to recognize objects if their size is *a priori* unknown.

To circumvent these difficulties, we have developed a mecha-

nism based on the *query by example* paradigm to recognize objects using a scale-invariant object model. The model is generated automatically from the example image by extracting the most salient image features and calculating their relative positions in a graph-like structure. The idea resembles the one formulated by Wiskott et al. [11] to recognize individual faces from images. However, our approach is suitable for *any object* at a broad scaling range and does not require image normalization or the selection of reference points by hand. To further increase its scaling range, our object model is easily expanded to a multi-resolution coding scheme.

Due to its generality our mechanism can be used for classical recognition and image understanding tasks such as scene decomposition, video object tracking, and even Optical Character Recognition (OCR).

2. A SCALE-INVARIANT OBJECT MODEL

The main idea of our approach to recognize objects in images is to model the object by its most salient features and search for matching object representations in the target image(s). Here, the object model is independent of the object's initial size, i.e. the recognition process corresponds to a scale-invariant matching task. The searched object itself is given by an example image, and the salient features of the object are located at the so-called *Points of Interest*. In fact, we apply a rigorous data compression to both the object and the target images, which guarantees an efficient recognition mechanism.

2.1 The Object Model

Suppose an object is given by an example image (the *object image*) such as Figure 1a, and should be recognized in a different image (the *search image*). Besides many other problems, there are two major difficulties: If the object is shown in the search image, its size might be unknown, and/or it might be partially occluded by other objects.

To recognize objects of arbitrary size, many mechanisms generate several scaled versions of the object image (i.e. images of the same object differing only in their size) and start a separate recognition process for each version. This computationally expensive procedure might not be suitable for real-time applications such as online database search.

In the presence of occlusion, the success of the recognition process depends on the used mechanism, the degree of occlusion, and the image data itself. Generally, it is not possible to predict whether an occluded object will be recognized or not. However, the *probability* of recognizing the object varies with the applied recognition strategy.

Our approach is based on the assumption that objects can be characterized by a few object-specific features which correspond to salient features in the object image. These features are used to generate a *model* of the object which is presented to the recognition process. As an example, Figure 1a shows the image of a stool. Some regions of the image (like the black background or the inner part of the seat) contain "empty" or "flat" areas. In contrary, the regions labeled with circles in Figure 1b show the details of the image which are "more important" for a representation of the stool. We call these regions containing salient image features the *Points of Interest* (POI).
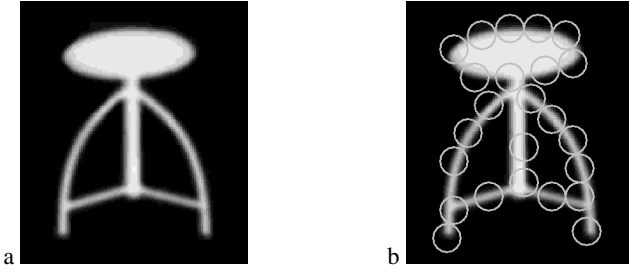
Figure 1: a) The image *Stool* and **b)** its Points of Interest.
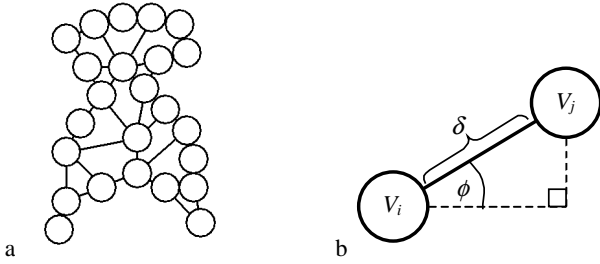

Figure 2: a) The graph-like model of Figure 1. For simplicity, only a subset of all edges is shown. **b)** Polar coordinates (ϕ, δ) of the relative position of two POI V_i, V_j.

To generate the object model, we first have to determine the Points of Interest of the object image (see the following sections). Next, the salient features and their relative positions are stored in a graph-like structure. Figure 2a shows the resulting model graph of Figure 1. Each node represents a POI and is labeled with the image feature at the corresponding position. The edges of the graph denote the relative positions of the POI in polar coordinates (ϕ, δ) where ϕ is the angle and δ the relative distance between two POI (see Figure 2b). Note that δ can be expressed in dimensionless units: For example, we could divide the absolute distance d of two POI by the average distance $\langle d \rangle$ of all the POI. Thus, the model is scale-invariant, i.e. the model is independent of the size of the underlying object.

To recognize the object in a search image, we have to look for image features which are similar to the salient features stored in the object model. Since the relative positions of the POI are known, the possible size and position of the object in the search image are predictable. Furthermore, if the object is partially occluded and only a subset of all the features of the model is detected, this information can still be used to decide whether the search image contains the object or not.

Thus, the scale-invariant object model provides a very efficient strategy for recognizing objects independently of their size, even if the object is partially occluded. However, we still have to determine the necessary Points of Interest and salient image features used to generate the model. The necessary techniques are discussed in the following sections.

2.2 Image Features and Image Primitives

Usually, every image consists of many image features such as lines, edges, or textures constituting the shown objects. Since these features provide a more abstract description of images than pixels they are often used for object recognition purposes.
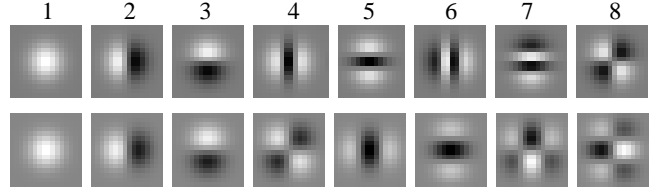

Figure 3: A typical set of GPCA primitives (*top row*) and the set of the first eight Scale Space kernels (*bottom row*). Due to experimental results, the first primitive of both sets is not used for recognition purposes (see section 4).

Image features are typically represented by the combination of a few atomic features (basis features or *image primitives*). For example, two-dimensional Gabor-Wavelets are often used as image primitives (see [11]). In general, image primitives should provide the information about the underlying image data in a compact representation to guarantee the efficient execution of the actual image processing task.

A classical technique which is based on the extraction of image primitives by explicitly utilizing the statistics of a given image is *Principal Component Analysis* (PCA). Subdividing an image into small image patches (*subimages*) and writing these patches as column vectors, the primitives are derived from the orthonormal eigenvectors (*eigenimages*) of the covariance matrix of the column vectors. Each subimage can be represented by a linear mixture of the eigenimages. The mixture components themselves are decorrelated – those mixture components with the highest variance are called *Principal Components*.

Since the reconstruction of the subimages from their Principal Components is optimal in the mean square error sense, the eigenimages should reflect most of the information contained in the image data. However, the associated mixture components are decorrelated but not statistically independent. Thus, we combined methods derived from *Independent Component Analysis* (ICA) [4] with PCA to obtain the *Principal Independent Component Analysis* (PICA) of an image [2]. The PICA components are as statistically independent as possible (i.e. their information content is as large as possible) and optimal in the mean square error sense.

Due to the rectangular subimages, the associated image primitives of both PCA and PICA are rectangular as well. Since this artificial shape may cause unwanted effects at the primitives' borders, we studied the PCA and PICA of subimages weighted with a two-dimensional Gaussian [3]. To our surprise, we found that the resulting GPCA[1] primitives contrary to the GPICA primitives do not significantly vary with the size of the underlying image, and that the same set of GPCA primitives is suitable even for different images. Furthermore, GPCA primitives can be calculated efficiently using a simple image model. Thus, GPCA primitives became first our choice to represent basic image features.

---

[1] GPCA / GPICA = Gaussian-weighted PCA / PICA. Note that GPCA or GPICA primitives are different from simple PCA or PICA primitives weighted with a two-dimensional Gaussian.

Figure 3 shows a typical set of GPCA primitives compared to kernels derived from Scale Space theory [7]. These kernels (*Scale Space primitives*) are the derivatives of a two-dimensional Gaussian and resemble the GPCA primitives (although they are not identical). Due to our experimental results described in section 4, we do not use the first primitive of either sets.

2.3 Image Encoding and Points of Interest

To describe a given image by a set of n image primitives, the image is convoluted with each primitive. This results in n filtered images F_i, $i \in \{1,\ldots,n\}$, where the number of coefficients per filtered image is roughly the same as the number of pixels in the original image. Thus, the coefficient $F_i(x, y)$ of the i th filtered image denotes the presence (or absence) of the i th primitive at the corresponding image position x in horizontal and y in vertical direction. The coefficients $F_i(x, y)$ *encode* the image with respect to the primitive set: They can be arrayed in n-dimensional column vectors or *feature jets*

$$J(x, y) = [F_1(x, y),\ldots, F_n(x, y)]^{\text{T}}$$

Each jet represents an individual image feature at position (x, y). Usually, the filtered images are *subsampled* to reduce the number of jets, i.e. only every m th jet in horizontal and vertical direction is used in subsequent processing stages.

However, to generate the object model described in section 2.1 above, we are only interested in the most salient image features (i.e. feature jets) located at the Points of Interest of an image. A simple approach to find the POI is presented by Itti and Koch [6]. Here, the POI are identified with the positions of isolated "peaks" in the filtered images. If such a peak appears simultaneously in some of the different filtered images at the same image position, this position is assumed to contain an important image feature and considered to be a potential candidate for a POI.

According to Itti and Koch, the POI are determined by selecting large coefficients of a *saliency map* denoting the locations of salient image regions. To calculate the saliency map S, the normalized absolute filtered images F_i are weighted and accumulated in S. We assume the mean $\langle F_i(x, y) \rangle$ of the filtered image coefficients $F_i(x, y)$ to be zero; if the mean is non-zero, we subtract it from the $F_i(x, y)$. The coefficients of the saliency map S are derived from the weighted sum

$$S(x, y) = \sum_i g_i \cdot \Gamma_i(x, y)$$

$$\text{where} \quad \Gamma_i(x, y) = \frac{|F_i(x, y)|}{max_{x,y}|F_i(x, y)|} \quad (1)$$

First, the absolute values of the coefficients of the filtered images are transformed to normalized coefficients $\Gamma_i(x, y) \in [0,1]$. Thus, coefficients $F_i(x, y)$ representing a high activity of the i th primitive at position (x, y) are mapped to coefficients $\Gamma_i(x, y)$ close to unity (note that $F_i(x, y)$ can be both positive or negative). Next, the resulting $\Gamma_i(x, y)$ are weighted by a constant weight factor g_i and summed up in the $S(x, y)$. To derive the weight factors, Itti and Koch used the squared difference $g_i = (M_i - \overline{m}_i)^2$ of the global maximum M_i and the mean $\overline{m}_i$ of the local maxima (peaks) of the $\Gamma_i(x, y)$. The idea is to emphasize those filtered images showing

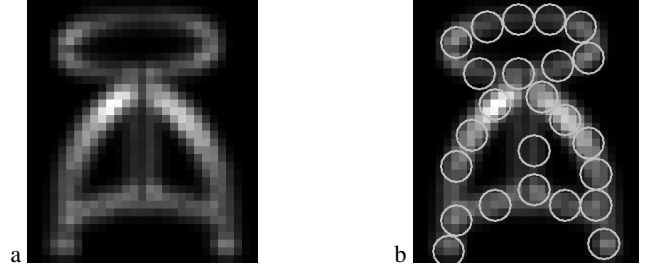**Figure 4:** **a)** The saliency map of the image in Figure 1a and **b)** the determined POI (same as in Figure 1b).

a few isolated but strong peaks which are supposed to represent salient image features.

However, the determination of the local maxima of a two-dimensional map $\Gamma_i(x, y)$ can be a very time-consuming task. Thus, we developed a different and more simple technique to calculate the weight factors g_i by using the *kurtosis kurt(F_i)* of the coefficients $F_i(x, y)$ [3]. Assuming the $F_i(x, y)$ to be samples of a random variable F_i with zero mean, the kurtosis $kurt(F_i)$ is derived from the fourth normalized central moment of F_i

$$kurt(F_i) = \frac{\left\langle \left(F_i(x, y) - \langle F_i(x, y) \rangle \right)^4 \right\rangle}{var(F_i)^2} - 3$$

$$= \frac{\left\langle F_i(x, y)^4 \right\rangle}{var(F_i)^2} - 3 \in [-3, \infty] \quad (2)$$

where $\langle . \rangle$ denotes expectation and $var(F_i)$ the variance of F_i. The kurtosis $kurt(F_i)$ delivers large positive values if only a few of the $F_i(x, y)$ deviate widely from their mean $\langle F_i(x, y) \rangle = 0$ and most of the $F_i(x, y)$ are close to zero. In contrary, $kurt(F_i)$ is small or negative if the $F_i(x, y)$ are clustered around zero but do not deviate widely. Thus, the kurtosis is a suitable measure to set the "importance level" of a filtered image during the generation of the saliency map in equation (1): Defining the weight factors by

$$g_i = kurt(F_i) + 3 \quad (3)$$

will emphasize those filtered images containing only a few but strong peaks as stated above.

Figure 4a shows the saliency map generated for the stool image in Figure 1a according to equation (1) using Scale Space primitives (see Figure 3). Bright dots represent image features with high saliency while dark dots represent "unimportant" features. To determine the POI shown in Figure 1b and Figure 4b from the saliency map, the locations (x, y) of coefficients $S(x, y)$ greater or equal to the mean $\langle S(x, y) \rangle$ of the map are chosen. Furthermore, we allow only one POI in a small circular region (typically of the same size as the image primitives). This enforces a minimum distance between the POI and prevents them from clustering at "hot spots".

2.4 The Multi-Resolution Object Model

Using the determined Points of Interest of an object image, the object model described in section 2.1 can be generated as follows: The feature jets at the POI are stored in the nodes of the
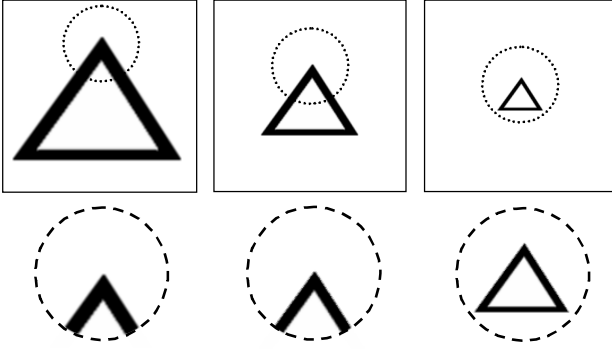
Figure 5: (*top row*) Object images differing in their size. (*bottom row*) Feature varying with the object's size.

object graph, while the edges hold the relative positions of the POI. Due to the notation of these relative positions in dimensionless polar coordinates, the object graph is scale-invariant, i.e. independent of the object's size. In contrary, the features still vary with the size of the object, as shown in Figure 5.

Here, the object, a triangle, is given by three images of different size (top row of Figure 5). Consider an image feature located at the upper corner of the triangles (denoted by a circle): Since the size of the support of the feature is constant, the feature's shape obviously varies with the scale of the object image (see bottom row of Figure 5). Thus, image features are in general *not* scale-invariant.

However, experimental results showed that the features remain almost constant for small changes in scale, while significant differences are caused only by broad variations of the object image's size. We found that valid scaling ranges [2] are $0.7 – 1.5$ for image features generated from GPCA primitives, and $0.5 – 2.0$ for features generated from Scale Space primitives (derivatives of the two-dimensional Gaussian function).

What do these results mean to the object model? If the size of the object shown in the search image lies within the valid scaling range or is approximately known a priori, the model can be used without modifications for recognition. In contrary, if the size is less than half or more than double the size of the searched object, it will not be recognized. In this case, we have to generate several object models from differently sized versions of the same object image; each of these models is valid for a specific scaling range. Thus, it is still possible to find arbitrarily sized instances of the object by performing a parallel search using the different models.

This multi-resolution approach based on the modeling of the object at different scaling stages seems to contradict the criticism stated at the beginning of section 2.1. However, since the valid scaling range of a single object model is rather large, only a few

---

[2] Here, the scale is measured by the ratio t (the *scaling factor*) of the width or height of the scaled image and the width or height of the original image. Thus, for $t < 1$ the scaled version will be smaller in size than the original image, and larger if $t > 1$, while $t = 1$ denotes no scaling at all. Note that the area of the scaled image varies with the squared scaling factor t^2.

models at different scaling stages are needed. Furthermore, the computational cost of the multi-stage recognition process is still acceptable even for real-time purposes.

3. THE RECOGNITION PROCESS

Given its multi-resolution model, an object is recognized by determining similar image features in the search image and verifying if their positions match the corresponding positions in the model. Thus, the search image has to be encoded as well, i.e. the image features of the search image must be calculated. However, neither the encoding of differently scaled versions nor a model of the search image are required. For database applications, the encoded search images can be calculated offline and stored in the same or a separate (meta) database to prevent computational overhead.

3.1 Identifying Similar Image Features

The determination of the image features of a search image is nearly the same as described in section 2.3 for object images. Again, we only use the "important" feature jets at positions (x, y), where the corresponding saliency coefficient $S(x, y)$ is greater or equal than the mean $\langle S(x, y) \rangle$ of the image's saliency map. The difference is, that the number of salient features per image region will not be restricted: we explicitly allow the clustering of features in the search image since we do not know a priori which of the features in the cluster might be the best match to a salient object feature.

Here, the similarity of feature jets is measured by their cosine, which in this case is equivalent to the correlation coefficient. Let $J(x, y) = [F_1(x, y),…, F_n(x, y)]^T$ be a feature jet of the object and $J'(x', y') = [F'_1(x', y'),…, F'_n(x', y')]^T$ be a feature jet of the search image. The *cosine* $cos[J(x, y), J'(x', y')]$ is defined by

$$cos[J(x, y), J'(x, y)] = \frac{\sum_i F_i(x, y) \cdot F_i'(x', y')}{\sqrt{\sum_i F_i(x, y)^2} \cdot \sqrt{\sum_i F_i'(x', y')^2}} \quad (4)$$

The cosine takes values from the interval $[–1,1]$: The closer the cosine to unity, the more similar are the two jets. A zero cosine corresponds to dissimilarity while a negative values close to $–1$ denotes that the first jet is similar to the inverted second jet. In our experiments, two feature jets were considered to be similar if their cosine was greater or equal than 0.9.

To simplify the identification of similar features, we use a straightforward labeling scheme. Every node V_i in the object graph is labeled with a unique index $i \in \{1,…, N\}$ where N is the total number of nodes. If the feature jet of the node V_i is similar to a jet at position $P = (x, y)$ in the search image, P is labeled with the index i:

$i \in label(P) \iff$ the jet of P is similar to the jet of V_i

Note that $label(P)$ represents a *set* of indices since the jet of P may be similar to more than one salient feature jet of the object. Figure 6 shows an example for features in a search image which are similar to a salient feature in the object image *Stool*.

3.2 Identifying Matching Edges

After the detection of similar features, we look for pairs of detected features in the search image matching the edges in the
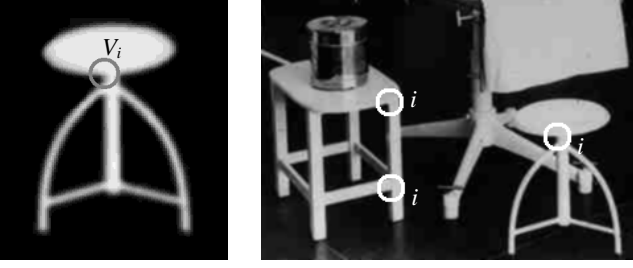
Figure 6: (*left*) Example for a salient object feature at V_i and (*right*) similar features in a search image.

Figure 7: (*left*) An edge (V_i, V_j) of the object model. (*right*) Matching edges (solid lines) and non-matching edges (dotted lines) in the search image. Points with features similar to the ones of V_i and V_j are labeled with i and j respectively.

object graphs at different scaling stages. This is done by comparing the feature labels and the relative positions of the jets.

To describe this process in detail, let V_i and V_j be two nodes in the object graph at a given scaling stage. The edge (V_i, V_j) is labeled with the relative position of V_i and V_j given by the angle ϕ and the distance δ. Since δ is the quotient of the absolute distance d and the average distance $\langle d \rangle$ of all the POI at the actual scaling stage, we have $d = \delta \cdot \langle d \rangle$.

Now, let P_1 and P_2 be the positions of two feature jets in the search image. Their relative position is given by the angle ϕ_P and the absolute distance d_P. The quadruple (P_1, P_2, i, j) is a *matching edge* to the edge (V_i, V_j) in the object graph (denoted by the relation operator $\rightarrow_m$), if all of the following four conditions are met:

$(P_1, P_2, i, j) \rightarrow_m (V_i, V_j) \Leftrightarrow$

   1) $i \in label(P_1)$, i.e. the jet of P_1 is similar to the jet of V_i

   2) $j \in label(P_2)$, i.e. the jet of P_2 is similar to the jet of V_j

   3) $\phi - \theta \leq \phi_P \leq \phi + \theta$

   4) $d_P = t_P \cdot d$,   where $t_{min} \leq t_P \leq t_{max}$
                      and $0 < t_{min} < 1 < t_{max} < \infty$

Thus, P_1 and P_2 have to contain features which are similar to the object features of V_i and V_j. Furthermore, the angle ϕ_P of the features' relative position is allowed to deviate up to a constant θ from the angle ϕ of the object features' relative position, while the absolute distance d_P of P_1, P_2 is allowed to be at least t_{min}-times and at most t_{max}-times the absolute distance d of V_i, V_j. The factor t_P is called the *scale* of the edge (P_1, P_2, i, j).

Typically, we set $\theta \approx 8°$, while t_{min} and t_{max} depend on the primitives used for encoding. As stated in section 2.4, the features of an image do not significantly vary within a scale of $0.7 \leq t \leq 1.5$ if GPCA primitives are used for encoding, and $0.5 \leq t \leq 2.0$ for Scale Space primitives. Consequently, we use these bounds to define t_{min} and t_{max}: Within the scaling range $t_{min} \leq t_P \leq t_{max}$ the image features can be compared to the salient object features, i.e. they may belong to a scaled version of the object in the search image we want to recognize.

In Figure 7, matching edges are represented by solid lines and non-matching edges by dotted lines. Obviously, the edge (V_i, V_j) in the object image is found twice in the search image. This is due to the tolerant matching process: We explicitly allow the matching of scaled and/or slightly rotated edges to facilitate the
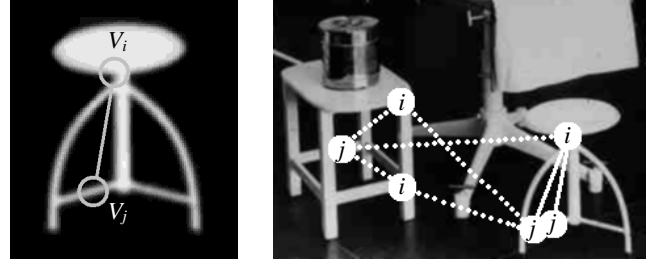
recognition of scaled and/or deformed object versions. However, edges that do not meet the conditions stated above are rejected.

3.3 Identifying Connected Matching Edges

In the next step of the recognition process, the matching edges are examined to find connected edges constituting a graph in the search image: Each of those graphs may represent a possible instance of the object.

Let (V_i, V_j), (V_k, V_l) be two arbitrary edges in the object image, and (P_1, P_2, i, j), (P_3, P_4, k, l) two of the corresponding matching edges in the search image, i.e. $(P_1, P_2, i, j) \rightarrow_m (V_i, V_j)$ and $(P_3, P_4, k, l) \rightarrow_m (V_k, V_l)$. How do we decide if the two matching edges are connected?

First, both edges must contain a node with the same position in the search image; this ensures the edges to be geometrically connected within the search image. Usually, we allow the nodes' positions to deviate from each other by a small amount Δ. For example, if the distance between P_2 and P_3 is less or equal to Δ, P_2 and P_3 are said to have the *same position* which is denoted by the relation $\sim_\Delta$, i.e. $P_2 \sim_\Delta P_3$.

Second, the features jets of nodes with the same position must be similar to same the feature jets in the object model. Remember that the nodes P_1 and P_2 of the edge (P_1, P_2, i, j) are labeled with the sets $label(P_1)$, $label(P_2)$ of indices indicating the similarity of their features to sets of corresponding object features. However, the nodes P_1 and P_2 are "bound" by the indices i and j to specific object nodes V_i and V_j, since (P_1, P_2, i, j) only matches the object edge (V_i, V_j). Of course, the same is true for the edges (P_3, P_4, k, l) and (V_k, V_l). Thus, if for example $P_2 \sim_\Delta P_3$, we require $j = k$.

Third, both edges must have a similar scale t_P. As seen in the previous section, the absolute distance d_P between the nodes of a matching edge is allowed to deviate from the absolute distance d between the nodes of the matched object edge by the scaling factor t_P: $d_P = t_P \cdot d$. If the two matching edges belong to the same graph in the search image, their scale has to be the same, or else the graph does not represent a valid instance of the searched object model. To distinguish between the scales of (P_1, P_2, i, j) and (P_3, P_4, k, l), we use the functions $scale[(P_1, P_2, i, j)]$ and $scale[(P_3, P_4, k, l)]$, respectively. In general, the scales of two matching edges never have exactly the same values. Thus, we divide the possible scaling range $[t_{min}, t_{max}]$ into m overlapping
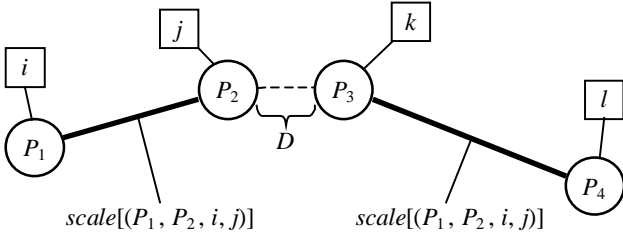
Figure 8: Example for the connection of matching edges. (P_1, P_2, i, j) and (P_3, P_4, k, l) are connected at the nodes P_2 and P_3, if $P_2 \sim_\Delta P_3$ (i.e. the distance D is smaller than Δ), P_2 and P_3 are indexed with the same node of the object model (i.e. $j = k$), and the scales of the edges are similar (i.e. $scale[(P_1, P_2, i, j)] \approx scale[(P_3, P_4, k, l)]$). If any of these conditions is not met, the edges are not connected at P_2, P_3.

Figure 9: a) The problem of *scattered nodes*. **b)** Geometrical interpretation of $scatter(\{P_v\}_j)$.

intervals $T_1, \ldots, T_m$. The scales of two edges are defined to be *similar* if there is an interval T_v where $scale[(P_1, P_2, i, j)] \in T_v$ and $scale[(P_3, P_4, k, l)] \in T_v$. We denote the similarity of scales by the operator $\approx$, i.e. $scale[(P_1, P_2, i, j)] \approx scale[(P_3, P_4, k, l)]$.

Thus, two matching edges (P_1, P_2, i, j) and (P_3, P_4, k, l) are said to be *connected*, if $scale[(P_1, P_2, i, j)] \approx scale[(P_3, P_4, k, l)]$, and at least one of the following four conditions holds:

1) $P_1 \sim_\Delta P_3$ *and* $i = k$, *or*

2) $P_1 \sim_\Delta P_4$ *and* $i = l$, *or*

3) $P_2 \sim_\Delta P_3$ *and* $j = k$, *or*

4) $P_2 \sim_\Delta P_4$ *and* $j = l$

Figure 8 shows an example for two connected matching edges.

3.4 Identifying Instances of the Object Model

The last step of the recognition process is the most crucial one. Having determined all the connected matching edges, we get several disjoint graphs in the search object. Each of these graphs is a possible instance of an object graph at a certain scaling stage. But how do we decide if the search image contains the object to be recognized?

In fact, the graphs are ordered according to a similarity measure $Similarity(G) \in [0,1]$ which denotes the *possibility* that a detected graph G represents the object in the search image. Depending on the current application, this measure is used to decide whether the recognition process was successful or not. For example, the search criteria could be relaxed to detect objects which are similar but not identical to the searched object. This is helpful in image retrieval tasks where the recognition system is used to filtered the huge amount of image data, and only a reasonable choice of matching images is presented to the user.

At the time this paper was written, we examined three criteria to calculate the similarity measure $Similarity(G)$ of a graph G:

1) The number $N_{Edge}(G)$ of matching edges in graph G,

2) the number $N_{POI}(G)$ of different POI of the object which are detected in G, and

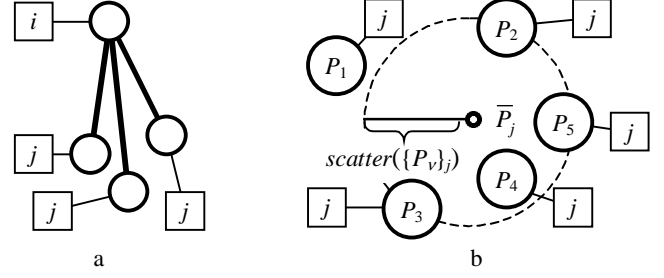3) the *average geometrical matching AGM(G)* of the detected POI in G.

The first criterion $N_{Edge}(G)$ is the most simple one: We just have to count the number of matching edges constituting the graph G. The higher the number of connected matching edges, the higher should be the possibility that G represents an instance of the object. However, our experimental results showed that this criterion is not suitable: Large graphs corresponding to incorrect object matches are preferred to small graphs representing the object correctly. Thus, we rejected $N_{POI}(G)$ as a measure for similarity.

The second criterion provides better results: $N_{POI}(G)$ counts the number of different POI of the object which are "detected" by G. To avoid the problem of *scattered nodes*, $N_{POI}(G)$ is utilized in conjunction with the third criterion, the *average geometrical matching* of detected POI in G. In the following, this procedure is described in detail.

Occasionally, some matching edges corresponding to the same edge (V_i, V_j) in the object graph are found, where the nodes with feature jets similar to the jet of V_j are scattered around the expected "correct" position of V_j (*scattered nodes*; see Figure 9a). This is possible since matching edges are allowed to be slightly rotated versions of edges of the object model. In general, scattered nodes do not influence the recognition process. However, in some cases they can lead to a serious problem where matching edges are arrayed according to a cascaded structure in the search image (see for example Figure 10). We could try to delete the surplus matching edges from G, but it would be difficult to decide which of these edges should be left, and how to deal with those edges in G that are connected to the deleted ones.

To solve this problem, we chose a simple heuristic approach. Let $POI(G)$ be the set of all indices j where the salient feature jet of a node V_j in the object graph is similar to the jet of a node in the graph G. Informally, $POI(G)$ are the indices of all the POI in the object image that are "detected" by the current graph G in the search image. Their number is given by $N_{POI}(G) = |POI(G)|$.

Now, let $\{P_v\}_j$ be the positions P_v of all nodes contained in the matching edges of G, where the corresponding feature jets are similar to the jet of a single node V_j, $j \in POI(G)$. We define the *geometrical scatter* of $\{P_v\}_j$ as follows:

$$scatter(\{P_v\}_j) = \frac{1}{|\{P_v\}_j|} \cdot \sqrt{\sum_{P_v \in \{P_v\}_j} (P_v - \overline{P}_j)^2} \qquad (5)$$

where $\overline{P}_j$ is the geometrical mean of the P_v and $|\{P_v\}_j|$ denotes the number of elements in $\{P_v\}_j$ (see Figure 9b). The *geometrical*
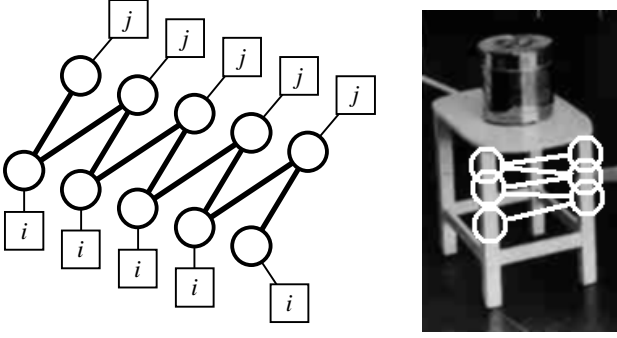
Figure 10: Cascaded matching edges caused by scattered nodes.

matching of $\{P_v\}_j$ is a simple transform of the geometrical scatter to values in the interval [0,1] based on a Gaussian function:

$$GM(\{P_v\}_j) \; = \; exp[-0.5 \cdot scatter(\{P_v\}_j)^2] \qquad (6)$$

$GM(\{P_v\}_j)$ is equal to unity, if the P_v do not scatter around their geometrical mean, and close to zero, if $scatter(\{P_v\}_j)$ becomes large. The *average geometrical matching* of the graph G is the expectation of $GM(\{P_v\}_j)$ over all detected POI:

$$AGM(G) \; = \; \frac{1}{N_{POI}(G)} \cdot \sum_{j \in POI(G)} GM(\{P_v\}_j) \qquad (7)$$

Thus, the average geometrical matching of the current graph G measures the "fitting" of the detected POI to their "correct" position as given by the object model (i.e. $AGM(G)$ is close to unity), or the "distortion" of G compared to the object graph at the actual scaling stage (i.e. $AGM(G)$ is close to zero).

According to these results we define the similarity measure for a graph G as follows:

$$Similarity(G) \; = \; c_1 \cdot \frac{N_{POI}(G)}{N} + c_2 \cdot AGM(G) \qquad (8)$$

where N is the number of all POI (or nodes V_j) in the object graph, $c_1 + c_2 = 1$ and $0 \le c_1, c_2 \le 1$; in our experiments we used $c_1 = c_2 = 0.5$.

4. EXPERIMENTAL RESULTS

Currently, the development of our mechanism for the scale-invariant recognition of objects in images is still in progress. However, we give some preliminary results to demonstrate the applicability of our approach.

We found that the best recognition results are obtained if the first primitive of the used primitive set (GPCA or Scale Space; see Figure 3) is omitted in the encoding stage. The first primitive is given by (or, in case of GPCA primitives, resembles) a two-dimensional Gaussian function, and encodes the average intensity of the support[3] of an image feature. Here, the average intensity represents the image's local brightness at the position of the current feature. Thus, our experiments indicate that the local

---

[3] Remember that the feature jets are derived by convoluting the primitives with the underlying image (see section 2.3). Thus, each feature is calculated for a small patch (subimage) of identical size as the primitives, the features' *support*.
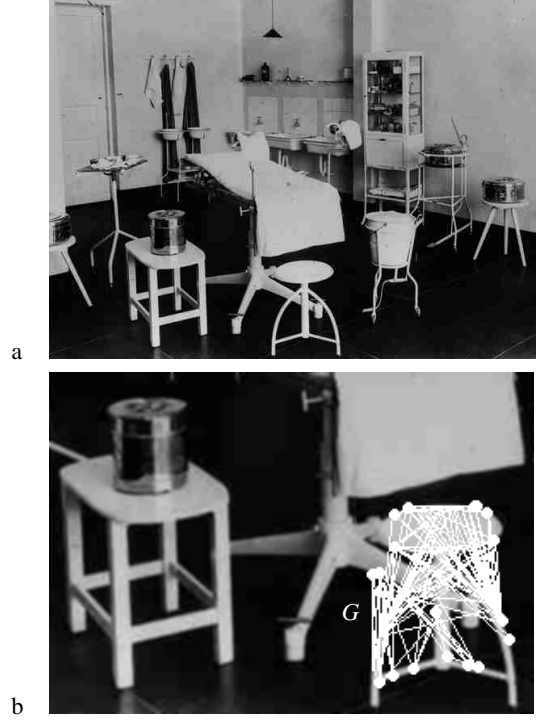

a

b

Figure 11: a) The image *Room*. **b)** The graph G with highest *Similarity*(G).

brightness should be ignored during the recognition process. In fact, this corresponds to the demand to recognize objects independently of their actual illumination and brightness.

First, we tested the system to detect the object *Stool* (Figure 1a; 130×150 pixels in size) in the image *Room* (Figure 11a; 603×446 pixels in size). To generate the feature jets of the images, both images were convoluted with 15 Scale Space primitives (16×16 pixels) and subsampled by a factor $m = 4$. Figure 11b shows the graph G with the highest value of *Similarity*(G) obtained from the recognition process. Note that the stool is recognized, although the background object interferes with the foreground. The experiment was successfully repeated with scaled and slightly rotated versions of *Stool*, as well as with other objects such as the chair seen at bottom left of the image *Room*.

Next, we used an image of NASA's Apollo 10 mission patch and a sketched version of that patch (Figure 12a+b; 173×200 pixels) to be recognized in a photo of the Apollo 10 press conference held in July 1969 (Figure 12c; 600×391 pixels). Again, both object images could be detected at different scales. In case of the sketched version, this is of particular interest, since only the outline of the emblem is shown in the object image. However, the same object image will also be found at various positions in the image *Room*, although the patch is not included therein. Obviously, the sketched patch version is an example for a "nonspecific" object image: It is too general to be identified with a particular real-world object.

5. CONCLUSION

In this paper, we presented an efficient mechanism for recognizing objects in digitized images. The object is given by an exam-
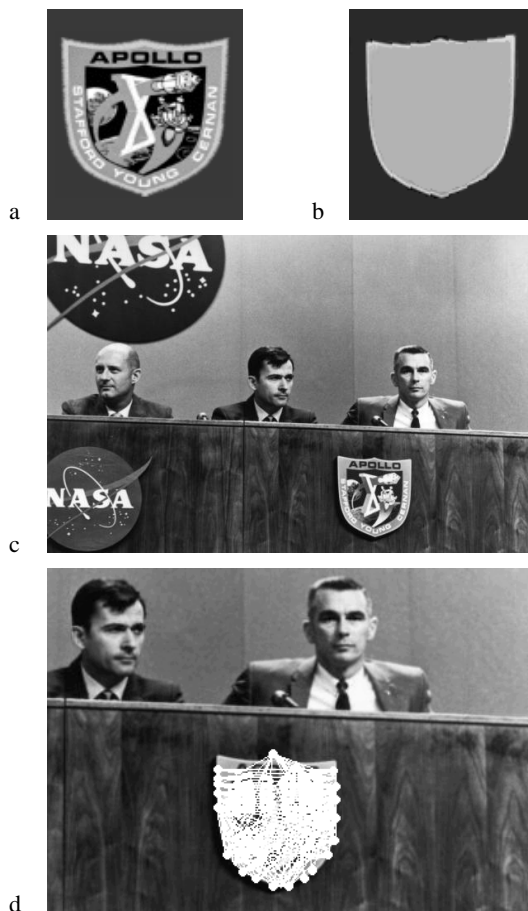
Figure 12: a) Apollo 10 mission patch (*source: NASA*). **b)** Sketched version of the patch. **c)** Photo from the Apollo 10 press conference (*source: NASA*). **d)** The first graph found for the sketched patch (b) in the photo (c).

ple image, while the recognition process itself is scale-invariant, i.e. independent from the object's size. Furthermore, the system is capable to detect objects in the presence of occlusion or slight distortion/rotation.

The recognition mechanism can be used in time-critical applications such as online image retrieval or video object tracing. The processing steps described in section 3 are explicitly designed for parallel execution. Thus, to further increase the performance of the presented algorithms, the mechanism can easily be implemented on a multi-processor system or realized as an all-hardware solution.

Currently, we are working on the improvement of the similarity measure described in section 3.4, which is used to compare the object model with possible instances of the object in the search image. Additional improvements and extensions of the mecha-

nism such as the application of color images, object detection by presenting object sketches, and 3D view-based recognition are planned and subject to future research.

A demonstration version of our mechanism (JAVA application) can be downloaded from the MASCOT homepage [1].

6. REFERENCES

[1] The MASCOT homepage.
http://seco.asa.cs.uni-frankfurt.de/Seco/mascot.html

[2] Arlt, B., Brause, R. The Principal Independent Components of Images. In Proceedings of ICANN'98 Vol.2 (Sweden, September 1998), Springer-Verlag.
extended online paper version:
http://www.cs.uni-frankfurt.de/fbreports/fbreport1-98.ps.gz

[3] Brause, R., Arlt, B., Tratar, E. Project SEMACODE: A Scale-invariant Object Recognition System for Content-based Queries in Image Databases. Internal report 11/99, CS Dept. (FB20), J. W. Goethe-University Frankfurt/Main, 1999.
http://www.cs.uni-frankfurt.de/fbreports/fbreport11-99.pdf

[4] Comon, P. Independent Component Analysis – A New Concept? Signal Processing 36/1994, 287-314.

[5] Flickner, M., et al. Query by Image and Video Content: The QBIC System. IEEE Computer, 23-32, 1995.
http:// wwwqbic.almaden.ibm.com/

[6] Itti, L., Koch, C. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), Vol. 20(11), 1254-1259, 1998.

[7] Lindeberg, T. Scale-Space Theory in Computer Vision. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[8] Ortega, M. et al. Supporting Similarity Queries in MARS. In Proceedings of MULTIMEDIA 1997, ACM Press, 403-414.

[9] Smith, J., Chang, S.-F. VisualSEEk: A Fully Automated Content-based Image Query System. In Proceedings of MULTIMEDIA 1996, ACM Press, 87-98.

[10] Virage, Inc. http://www.virage.com/

[11] Wiskott, L., et al. Face Recognition by Elastic Bunch Graph Matching. In Jain, L.C., et al. (eds.). Intelligent Biometric Techniques in Fingerprint and Face Recognition, Springer-Verlag, 1999