Accuracy versus speed in context-based object detection

Niek Bergboer *, Eric Postma, Jaap van den Herik

Micc-Ikat, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Available online 18 September 2006

Abstract

The visual detection and recognition of objects is facilitated by context. This paper studies two types of learning methods for realizing context-based object detection in paintings. The first method is called the gradient method; it learns to transform the spatial context into a gradient towards the object. The second method, the context-detection method, learns to detect image regions that are likely to contain objects. The accuracy and speed of both methods are evaluated on a face-detection task involving natural and painted faces in a wide variety of contexts. The experimental results show that the gradient method enhances accuracy at the cost of computational speed, whereas the context-detection method optimises speed at the cost of accuracy. The different results of both methods are argued to arise from the different ways in which the methods trade-off accuracy and speed. We conclude that both the gradient method and the context-detection method can be applied to reliable and fast object detection in paintings and that the choice for either method depends on the application and user constraints.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Computer vision; Object detection; Face detection; Machine learning

1. Introduction

The perception of visual scenes relies on two main kinds of information: the gist and the layout (Coren et al., 2004). The gist of a visual scene is the meaning of the objects and their relations constituting the scene. The layout of a scene refers to the spatial arrangement of objects. In visual arts, the automatic recognition of the gist of a painting is quite a challenging task because it requires an understanding of the semantics of the painted objects and scenes which is beyond the capability of present pattern-recognition systems (van den Herik and Postma, 2000). In contrast, the automatic recognition of the layout of a painting is feasible as it requires the detection of objects which is within reach of present-day techniques (Bergboer et al., 2004; Torralba, 2002).

In this paper, we constrain ourselves to the task of detecting objects using the local context. We define the local context as incorporating layout information from the object itself and its immediate spatial surroundings. In our previous work we have shown that layout information enhances the attentional selection of objects (Bergboer et al., 2004). In this paper we investigate two main learning methods for using context to detect objects. Both methods extract contextual information to estimate the location of an object, but differ in the way the estimation is obtained. The first method is called the gradient method. It learns to transform the spatial context into a gradient that points towards the object. The second method is called the context-detection method which is trained to detect image regions that are likely to contain an object. The aim of our study is to assess the performances of both methods in terms of the accuracy and speed of detection. For this purpose, both methods are applied to a face-detection task involving a large collection of images containing one or more faces in a variety of contexts.

The outline of this paper is as follows: Section 2 describes the gradient method, and Section 3 describes the context-detection method. The experimental setup is outlined in Section 4. Then, in Section 5, the results of both methods are presented. Section 6 shows an application in the domain of visual arts. Section 7 discusses the results. Finally, Section 8 concludes on the application of both methods in the domain of visual arts.
2. The gradient method

The gradient method takes selected regions of an image to estimate the location of the nearest object. The regions are called context windows. They are selected in the vicinity of an object or even at the location of an object. Hence, the gradient method relies on the “object-is-near” assumption stating that an object is present in the immediate vicinity of or even at the examined location. The square depicted in the left panel of Fig. 1 represents a context window. The arrow (pointing from the upper left corner of the square towards the centre of the contour lines) represents the gradient vector which is derived from the contents of the context window. In our experiments, we investigate two context-window sizes: small and large context windows. Small context windows are about the size of a face, whereas large context windows encompass a region twice as large as a face. Fig. 2 shows examples of both context-window sizes.

The gradient method consists of four stages. In the first stage, the image is preprocessed using a biologically plausible transformation. The second stage involves the local context-based estimation of object locations. In the third stage, the local estimates are integrated into a global saliency map. Finally, in the fourth stage, objects are detected. Below, we discuss each of these stages in detail.

2.1. Preprocessing

In the preprocessing stage of the gradient method, the contents of context windows is transformed into feature vectors by means of a standard multi-scale wavelet transformation. In the experiments reported here we use an overcomplete Haar-wavelet basis for transforming the image contents into a feature vector (Papageorgiou and Poggio, 2000).

We extract quadruple-resolution wavelets at wavelet scales 2 and 4 (i.e., the detail coefficients at decomposition levels 1 and 2) from context windows of size $19 \times 19$ pixels. In this way, relations between large-scale features on a fine spatial resolution are incorporated in the feature vector representation. Quadruple-resolution scale-4 wavelets yield $17 \times 17$ coefficients for a given orientation for a context window of size $19 \times 19$. As we use three orientations (horizontal, vertical, and diagonal), and two scales, it yields 1734-dimensional feature vectors for each context window. To facilitate further processing, the raw feature vectors are projected onto 16-dimensional reduced feature vectors $v$ using principal component analysis.

2.2. Gradient estimation towards object locations

In the second stage of the gradient method the feature vectors (representing the contents of the context windows) are transformed into estimates of object locations. The aim is to estimate the location of the object relative to the context window, i.e., to estimate the gradient vector $\overrightarrow{c}_{r} = (x_r, y_r)$ by means of the window features $v$, where $x_r$ represents the horizontal relative location and $y_r$ represents the vertical relative location. The transformation is learned using a training set consisting of the estimated $\overrightarrow{c}_{r}$ (acquired from context windows) and the associated true object position $\overrightarrow{x}_{r}$. As a learning algorithm we use cluster-weighted modelling (Gershenfeld, 1999), because it is straightforward and efficient.

The function that minimises the mean square error between $\overrightarrow{c}_{r}$ and $\overrightarrow{x}_{r}$ is the conditional expected value (Montgomery and Runger, 1994, p. 247):

$$\overrightarrow{x}_{r} = \int \overrightarrow{x}_{r} f(\overrightarrow{x}_{r}, v) \, d\overrightarrow{x}_{r},$$

(1)
where the joint probability density function (PDF) \( f(\vec{x}_r, v) \) describes the relation between the two random variables \( \vec{x}_r \) and \( v \). It is given by

\[
f(\vec{x}_r|v) = \frac{f(\vec{x}_r, v)}{f(v)}.
\] (2)

After training, the preprocessed contents of context windows is translated into a PDF for the relative object location. The left panel of Fig. 1 is an illustration of such a PDF (the contour lines). The arrow represents the estimate of the object location relative to the context window (the square). Fig. 2 illustrates the position and the extent of typical small and large context windows for the object class of faces. The large dotted squares denote the region from which the context windows (training samples) are taken, the top left corner of the dashed square in the centre corresponds to the true face location, and the solid square in the upper left corner of the training region represent the context window for a relative displacement of \(-8\) pixels in both the horizontal and vertical direction in the down-sampled image.\(^2\)

To achieve reliable context-based estimates of object location the cluster-weighted model is trained on examples of contexts of objects (i.e., frontal faces). The main design parameter for the cluster-weighted model is the number of clusters that are used. We train a model using eight clusters, as preliminary results have shown that further increasing the number of clusters does not improve detection. The algorithm is trained on a dataset of 1885 faces. Within the facial vicinity shown in Fig. 2, training samples are obtained from context windows at the relative displacements \(-8\), \(-6\), \(-4\), \(-2\), \(0\), \(+2\), \(+4\), \(+6\), \(+8\) in both the horizontal and vertical directions. Thus, 81 relative displacements for each face are used, which yields a total of 152,685 samples in the training set.

Fig. 3 shows examples of the cluster centres obtained after training the cluster-weighted model with small context windows (left) and large context windows (right). Each cluster has a centre in the input space (i.e., a combination of visual features), that is linked to a centre in the output space (i.e., a relative face location). Each row in the left half of the Figure corresponds to a cluster centre in the input space. The three columns show horizontal, vertical, and diagonal wavelet details, respectively. The grey value represents the magnitude of the brightness gradient in the given orientation. Note that the raw feature vectors contain samples at both scale 2 and scale 4. However, as the cluster centres at these two scales are rather similar, we have chosen to display the centres for scale 4 only. The cluster centres in the output space are superimposed on the images; each ellipse shows a one-standard-deviation confidence interval of the relative location with respect to the location of the face (indicated by an X). For instance, the pattern of brightness gradient magnitudes given in the top row is indicative of a position slightly to the upper-right of the actual face location.

Fig. 3 reveals that spatially large features, such as the eyes, the nose, the mouth, and the edges of the head, are used to locate the face. For instance, the second row of the left part of Fig. 3 shows that the presence of the eyes in the centre of the window, indicated by the two bright spots in the vertical details, provides a strong contextual clue that the current position is slightly above the location of the face. However, in practice, there will not be merely one cluster that determines the location of the face; the perceived features are projected onto a linear combination of the eight cluster centres, and the estimated face location will also be a linear combination of the relative positions of each of the clusters.

2.3. Integration of local estimates

The third stage of the gradient method is the addition of the PDFs obtained at all locations and scales in the image to yield a global “object saliency” map. The integration is illustrated in the middle panel of Fig. 1 for a given scale and two locations. PDFs are obtained at a grid of window locations. In order to obtain a larger-scale saliency map for the object, all individual PDFs are added. This adding process is performed by moving each individual PDF to its
absolute location in the image, which is obtained by adding the expectation value of the relative location to the current location of the window. In Fig. 3, the expectation value for the position based on features from the leftmost context window lies slightly below and to the right of the current location. The expectation value for the position based on features from the rightmost context window lies above and to the left of the current position. The integration stage results in an image-wide object saliency map. The right panel of Fig. 1 is an illustration of the object saliency map. The contour lines demarcate the saliency.

2.4. Object detection

In the fourth stage of the gradient method, objects are detected in regions that have a high-object saliency. At each scale, all locations are sorted by their object saliency, after which an object detector is employed at a given fraction of the locations with the highest saliency. We use a standard object detector by Viola and Jones (2001) and Lienhart et al. (2002), discussed in more detail in Section 3.1.

3. The context-detection method

The context-detection method differs from the gradient method in that it relies on the detection of regions that are likely to contain objects, rather than determining the gradient towards the object location. The context-detection method consists of two stages: a context-detection stage and an object-detection stage. Below, we discuss each of these stages in detail.

3.1. Context detection

In the context-detection stage an image is searched at a coarse scale for likely spatial contexts of the object class of interest. The context-detection stage is illustrated in the left panel of Fig. 4. The bold square represents the search window at a given location. The normal squares represent detected contexts. It results in the selection of a number of context regions. The search for likely contexts proceeds using a window-sliding technique which is widely used in visual detection methods, e.g., Sung and Poggio (1998), Schneiderman and Kanade (2000), and Papageorgiou and Poggio (2000). In window sliding techniques, a square window of a given size is slid over the image at different scales. At each position and scale, the contents of the window is used for deciding if it represents a likely object context.

In the context-detection stage we define an object’s spatial context as a square region of the image, of which the lateral dimensions are larger than those of the square region enclosing the object. We investigate three context sizes. The first context size is named “PPcon”, as it comprises a region twice the lateral size of a face label as defined by Papageorgiou and Poggio (2000). The second context size is named “VJcon”, as it comprises a region twice the lateral size of a face label as defined by Viola and Jones (2001). The third context size is named “VJ2con”, as it comprises a region that is laterally twice as large as the “VJcon” region. The three context sizes are illustrated in Fig. 5.

In the context-detection stage the contents of context windows is transformed into feature vectors by means of a standard multi-scale wavelet transformation. Instead of a separate Haar-wavelet basis that is employed in the gradient method, the context detection relies on a related and very efficient technique: the classifier by Viola and Jones (2001). This classifier combines a good classification performance with computational efficiency by employing a boosted cascade of simple classifiers that is trained using AdaBoost (Freund and Shapire, 1996), and for which simple Haar wavelets are used as features (Papageorgiou and Poggio, 2000). In our experiments we use the algorithm developed by Lienhart et al. (2002) who extended the Viola–Jones classifier and incorporated it into the publicly-available Intel Open Computer Vision Library.3

3.2. Object detection

In the object-detection stage objects are searched for and detected at a fine scale within the centre parts of selected context regions only. The object-detection stage is illustrated in the centre panel of Fig. 4. Here, the bold square with the array represents an object-detection window. The regions outlined by solid lines are the union of all detected contexts. The dashed regions represent the centre parts of the contexts that are searched by the object detector. The right panel of the Figure is the final state of the second stage; it shows the final detection result. We employ the same object detector as in the gradient method; it is described in the previous section.

3 http://www.sourceforge.net
Within detected context regions, the object classifier examines all location-scale combinations that lie within a factor 1.5 of the expected scale of a face. The expected scale depends on the type of the context window. For instance, for the VJcon context, a face is expected to be half the size of the context.

4. Experimental setup

In this section, we outline our experimental setup for both the gradient method and the context-detection method. In the next subsections, we discuss the data on which the experiments are based, the experimental methodology and the performance criteria that we use.

4.1. Data

In order to assess the performance of both methods, a total of 3306 frontal faces was labelled in 995 natural images. This dataset is divided into a test set of 775 images that together contain 1885 labelled faces, and a training set of 220 images with 1421 labelled faces. The test set contains labelled faces that are at least $30 \times 30$ pixels in size. To ensure that all faces are found, the images are classified for face sizes of $24.5 \times 24.5$ pixels and up. To this end, a scale-space pyramid of the image is calculated in which subsequent scales differ by a factor 1.1.

4.2. Experimental methodology

To assess the generalisation performance of the gradient methods, we employ 10-fold cross validation (Duda et al., 2001).

For the context-detection method, we refrained from using a cross-validation procedure. Instead, we used a single partitioning of the dataset into a training and set set. Because of the large size of both sets, we expected to obtain a reliable estimate of the generalisation performance. This expectation was confirmed in a preliminary experiment performed on a smaller subset of the data.

For the context-detection method, the three context classifiers—one for each of the context-types shown in Fig. 5—are trained on the contexts of the 1421 faces in the training set. A window size of $20 \times 20$ pixels is used, and 3000 negative examples are used in the training of each of the 20 classifier stages. Each of the stages is trained to have a detection-rate of 0.995 and a false-detection rate of 0.5.

The contextual selection, as specified above for both methods, corresponds to stages 1–3 of the gradient method, and stage 2 for the context-detection method. In the final stage of both methods (the object detection), we use the standard Viola–Jones frontal face-detector that is packaged with the Open Computer Vision Library, where, for accuracy reasons, an offset of 5 is added to the threshold of the final stage.

In the gradient methods, the trained models are used at different “step sizes”; for a given “step size” $s$, a PDF is calculated only after every $s$ pixels in both the horizontal and vertical direction. In the context-detection methods, a window-sliding technique is used for the context-detectors.

4.3. Performance criteria

To assess the performances of both methods, we employ three performance measures: true positives (i.e., correctly detected faces), false negatives (i.e., falsely rejected faces), and false positives (i.e., falsely detected faces).

From all object windows that are classified positively, at least one should overlap sufficiently with a labelled face in order for that face to be a true positive; the overlap criteria are: (1) the size of the window should be within a factor 1.5 of the size of the labelled face, and (2) the window’s centre should be within a distance to the labelled face’s centre that is not larger than 30 per cent of that face’s size. If none of the detected windows satisfies the two overlap criteria for a given labelled face, than the labelled face is a false negative. Detected windows that do not satisfy the overlap criteria for any of the labelled faces are regarded as false positives.

5. Results

As the aim of this study is to evaluate the accuracy and speed of context-based object detection methods we present their performances in terms of (1) Receiver Operating Characteristic (ROC) curves (expressing the detection rate as a function of the false positive rate) and (2) tables of detection times. By presenting the results in this way, the accuracies and speeds of both methods (and their variations) are emphasised and can be readily compared. Below we present comparisons of the detection accuracy (in 5.1) and of the detection speed (in 5.2).

5.1. Detection accuracy

To assess the benefits of the gradient method and context-detection method, the results obtained are compared to results obtained without a selection mechanism. For this purpose, all images were scanned pixel-wise in their entirety.
using the Open Computer Vision Library frontal face
detector, henceforth referred to as the brute-force method.

From Fig. 6 we clearly see that the small-window gradient
method outperforms the brute-force method (see the top and
middle panel of Fig. 6). For instance, at a detection rate
of 80%, the brute-force method yields 0.48 false positives per
image, whereas the small-window gradient method yields
0.22 false detections per image, which is a factor 2.16 lower
than the false-detection rate of the brute-force method.
The large-window gradient methods yields 0.47 false posi-
tives per image, which is similar to the brute-force method.

The top panel of Fig. 6 shows the ROC curves for the best-
performing gradient methods and the best-performing
context-detection methods in comparison with the brute-force
method. The centre panel of Fig. 6 shows the ROC curves
for the small-window and large-window gradient method
(the “combined gradient” method will be discussed in Sec-

tion 7). The bottom panel of Fig. 6 shows the ROC curves
for the three window sizes of the context-detection method.
From these results on the accuracy of context-based object
detection we derive the following two observations.

The first observation is that the best gradient method
outperforms the best context-detection method in detection
accuracy. It is clear from the top panel of Fig. 6 that the
small-window gradient methods yields significantly less
false detections at a given detection rate than the PPcon
context-detection method. The second observation is that
the small-window gradient method outperforms the large-
window gradient method. This is clearly visible in the
upper two plots shown in Fig. 6. For the context-detection
method the same observation can be made, although for
this method it should be remarked that both window sizes
perform worse than the brute-force method.

5.2. Detection speed

Tables 1 and 2 display the detection speeds for the
gradient and context-detection methods, respectively. Our
observations from the results on the detection speed of
context-based object detection are 3-fold.

First, the gradient methods are slower than the context-
detection methods. Second, the large-window gradient
method is faster than the small-window gradient method.
Third, among the context-detection methods, the small-
window PPcon performs best.
6. Application of the gradient methods to paintings

To evaluate the performance of context-based object detection in paintings, we apply the gradient methods and the context-detection methods to 53 selected paintings containing 57 labelled frontal faces. Fig. 7 shows the results of the brute-force method, the best performing gradient method (small-window with step-size 4), and the best performing context-detection method (PPcon).

In agreement with the results obtained on the natural-image dataset, the small-window gradient method performs best. In contrast to the results obtained on the natural-image dataset, the PPcon context-detection method performs slightly better than the brute-force method. The improved performance of context-based object detection in paintings as compared to natural images will be discussed in Section 7.

Fig. 8 shows a typical result obtained with the gradient method. The image, a self-portrait of Rembrandt van Rijn painted in 1669, is shown on the left. The centre image shows the added PDFs obtained in the third stage of the gradient method. Gray values represent probabilities ranging from black (low probability) towards white (high probability). The right image displays the final detection result: the solid squares represent object detections found in the first $10^{-3}$ fraction of the search space, whereas the dashed squares represent object detections found in the remainder of the search space.

### Table 2
Comparison of times required context selection of the gradient methods, for different image sizes

<table>
<thead>
<tr>
<th>Image size (pixels)</th>
<th>Brute-force time (ms)</th>
<th>Gradient method</th>
<th>Context time (s)</th>
<th>Speedup factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>800 × 600</td>
<td>1678</td>
<td>Small window</td>
<td>38.9</td>
<td>$4.3 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Large window</td>
<td>7.04</td>
<td>$2.4 \times 10^{-2}$</td>
</tr>
<tr>
<td>640 × 480</td>
<td>1124</td>
<td>Small window</td>
<td>24.0</td>
<td>$4.7 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Large window</td>
<td>5.20</td>
<td>$2.2 \times 10^{-1}$</td>
</tr>
<tr>
<td>320 × 240</td>
<td>259</td>
<td>Small window</td>
<td>8.99</td>
<td>$2.9 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Large window</td>
<td>2.38</td>
<td>$1.1 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

7. Discussion

Our results obtained with the two context-based object-detection methods gives rise to three considerations. The first consideration is related to the observations on the effect of window size in both the gradient and the context-detection methods. The second consideration concerns the “object-is-near” assumption that underlies the gradient method. Finally, the third consideration is about the global pattern of speed and accuracy performances obtained with the two methods. Below, we discuss each of these considerations.

#### 7.1. Small versus large windows

In both the gradient and context-detection methods, we observed that small context windows outperform large
context windows on detection accuracy. We analyzed the features associated with both window sizes to investigate the different results. Fig. 9 depicts typical PCA components obtained with the small-window (top row) and large-window context-detection methods. In general, the small-window components reveal more object-like shapes and patterns than the large-window components. For instance, in the first column the small-window component (top) contains a small circle against a larger white–black transition, whereas the large-window component (bottom) contains a white–black transition only. The occurrence of small details in the small-window components enhances the ability of the associated classifier to detect likely object contexts. Due to their size, the large-window components tend to reflect the principal components of natural images which look like Gabor filters (Baddeley and Hancock, 1992). Apparently, the small-window components are more descriptive for our detection task than the large-window components.

Regarding detection speed, it is expected that the use of small windows would worsen performance; for small windows, the image has to be analyzed down to smaller scales, which involves a larger number of model evaluations. For the gradient method, this is indeed the case. However, the context-detection methods using small windows are faster than those using large windows, despite the larger number of model evaluations. The reason for this seems to be the larger intra-class variance for larger windows: it increases the amount of features in the detection cascade, which slows down the execution speed. Apparently, the advantage of having a less complex context classifier outweighs the disadvantage of having to search down to smaller scales.

7.2. The “object-is-near” assumption

The basic assumption underlying the gradient method is that an “object is near to the current location”. Our “object-is-near” assumption may give rise to a high false-positive rate because if an object is not near to the current location, the method is likely to overestimate the probability of an object being near. To investigate whether this is the case, we combined the small-window version and large-window version of the gradient method by approximating the probability that an object is present, \( P(\text{object}) \) in the following way:

\[
P(\text{object}) \approx P(\text{object|object in vicinity})P(\text{context|context in vicinity}). \tag{3}
\]

The centre panel of Fig. 6 shows the ROC curves obtained with the gradient method using Eq. (3)—labelled “combined gradient method” and the original gradient methods. Clearly, the performance is worse. Analysis revealed that the application of Eq. (3) did lower the probabilities in locations where no objects were near. However, it also lowered the probabilities in the vicinity of objects. Hence, the overall performance worsened as compared to the original gradient method. It may be possible to attenuate the probability in another way to lower the false-positive rate of the gradient method, e.g., by estimating a low-detail image-wide PDF for likely object locations using the method of Torralba (2002). However, we believe that our “object-is-near” assumption does not harm the accuracy too much.

7.3. The accuracy-speed trade-off

The different patterns of results obtained with the two methods arise from the different ways in which both methods trade-off accuracy and speed. The gradient method emphasises accuracy by taking multiple independent samples from the context to estimate the gradient towards the object. The enhanced accuracy is costly in computational resources as is evident from the speed results. In contrast, the context-detection method is very fast for two reasons. The first reason is that it operates on a larger scale. The second reason is that it relies on a Viola–Jones implementation that is efficient. However, the enhanced speed comes at the cost of accuracy. Both methods are characterised by a different trade-off between accuracy and speed. We believe that yet another trade-off between accuracy and speed is obtained by developing a method that combines features of the gradient and context-detection methods.

The comparison between the gradient method and the context-detection method is currently performed by comparing a 10-fold cross-validated gradient method to a context-detection method trained on a single training-test set combination. The main reason for not using 10-fold cross validation for the context detection method was that training all boosted cascade classifiers for the context-detection method is impractical with our current computational resources. Admittingly, the use of two different validation methods may have affected our results. Therefore, we confirmed our results by ensuring that our training set formed a representative sample of the entire dataset. We did so by training a context detector on all 3306 faces in the dataset. This yielded the same qualitative results as obtained above. Apparently, the separate training set is representative of
the object class. Therefore, we are confident that our results are reliable estimates of the generalisation performances.

7.4. Generalisation to paintings

The results obtained on the dataset of paintings show a accuracy-speed trade-off similar to the trade-off in the natural-image experiment results; the small-window gradient method is more accurate than the PPcon method, and this accuracy comes at the cost of speed.

Strikingly, both context-based methods outperform the brute-force method in detecting painted faces. A possible reason for the superior performance on painted faces as compared to natural faces, is that painters deliberately emphasise objects (such as faces) by enhancing (or even exaggerating) the contrast with the background. For instance, by enhancing the differences in spatial frequency contents between object and background (see van Dantzig, 1973, for other examples). Apparently, the enhanced contrast explains the improved performance of our methods on painted faces as compared to natural faces.

8. Conclusion and future work

We conclude that both the gradient methods and context-detection methods can be applied to reliable and fast object detection in natural and painted images. The choice for either method depends on the application and user constraints. In our future work we intend to develop and evaluate context-based object-detection methods that combine features of both methods.

Acknowledgement

This research is carried out within the ToKeN project EIDETIC (grant number 634.000.001) of the Netherlands Organisation for Scientific Research (NWO).

References