

# The Role of Diffusion in Figure Hunt Games

Julia Diebold · Sibel Tari · Daniel Cremers

Received: 21 October 2013 / Accepted: 17 November 2014  
© Springer Science+Business Media New York 2015

**Abstract** We consider the task of tracing out target figures hidden in teeming figure pictures known as *figure hunt games*. Figure hunt games are a popular genre of visual puzzles; a timeless classic for children, artists and cognitive scientists. We argue and experimentally demonstrate that diffusion is a key to algorithmically search for a target figure in a binary line drawing. Particularly suited to the considered task, we propose a diffuse representation which diffuses the image while retaining the contour information.

**Keywords** Screened Poisson PDE and variants · Level sets · Non-linear diffusion · Figure hunt games · Teeming figure pictures · Applications of variational and PDE methods

## 1 Introduction

In 1986, the British illustrator Martin Handford created the distinctive red-and-white dressed character Wally. Since that day, *Where's Wally?* became an extremely popular series of children's books consisting of diverse illustrations, depicting dozens of people. Readers are challenged to find Wally in illustrations where an abundant number of small figures including Wally are brought together.

*Where's Wally?* is only a sample, though the most famous, in a popular genre of visual puzzles called *figure hunt games*. Figure hunt games have been a timeless classic for children, artists and cognitive scientists. As early as 1926 Kurt Gottschaldt experimented with intentionally designed hidden figures—simple drawings where simple shapes such as polygons are embedded within more complex organizations—to study the influence of experience on perception and the extent to which holes influence the perception of parts [10]. Gottschaldt type puzzles (Fig. 1 top row) together with the *Where's Wally?* type ones (Fig. 1 bottom row) form the focus of this paper. This sub-genre of the figure hunt can be generalized based on two factors as exemplified in Fig. 1. The first factor is the co-dimension: The individual figures in the top-row illustrations are one-dimensional objects drawn on top of each other whereas the ones in the bottom-row illustrations are two dimensional. The second factor is the number of targets. In the first column, each illustration contains a single target, whereas in the second column several targets (hangers and bees, respectively) are placed among distractors.

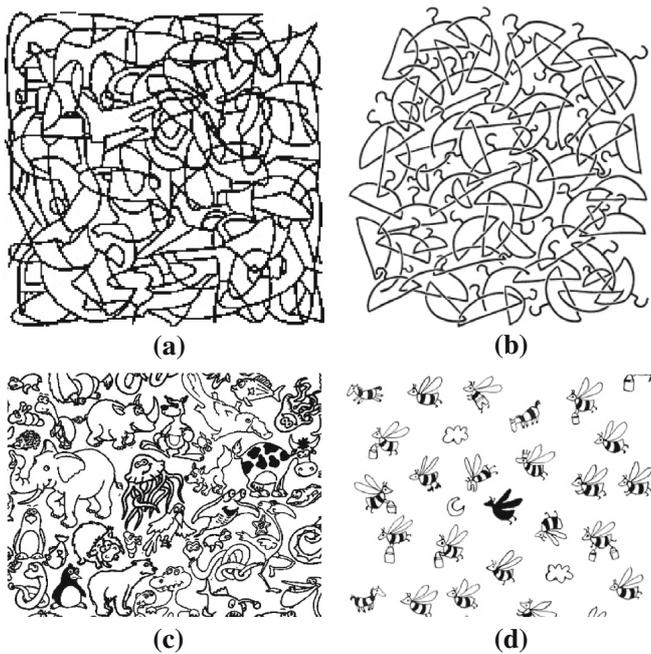
In the course of this paper, we will discuss how a computer program can trace out the contours of the hidden clover or hangers and locate the desired animals. The question might for example be: "Can you trace out the hidden cloverleaf?" or "How many hangers are there?", "Where is the sea star?" or "Are there other animals hidden among the bees?". We shall adopt the most straightforward solution: searching the entire illustration; hypothesize a pose and scale for the target figure and measure how well it fits. Among other things, this requires (1) a randomized search algorithm that will return multiple answers; (2) a fitness function which distinguishes a bad fit from a good fit. We will argue and experimentally demonstrate that replacing illustrations and/or target figures with diffuse forms significantly helps. Thus, the paper shows the effect of diffusion for the particular considered task of

---

J. Diebold (✉) · D. Cremers  
Technische Universität München, Munich, Germany  
e-mail: julia.diebold@tum.de

D. Cremers  
e-mail: cremers@tum.de

S. Tari  
Middle East Technical University, Ankara, Turkey  
e-mail: stari@metu.edu.tr



**Fig. 1** Samples of figure hunt games. Games with a single target (*left column*) versus several targets (*right column*) and Gottschaldt type puzzles (*top row*) versus *Where's Wally?* type ones (*bottom row*). **a** Hidden cloverleaf. **b** Several hangers. **c** Hidden elephant. **d** Multiple bees

figure hunt games. A secondary issue is how to speed up the search process. To this end, we will experiment with a coarse-to-fine strategy using diffuse forms.

A preliminary conference version examining only the Gottschaldt type puzzles has appeared in SSVM 2013 [6]. In this paper, we concentrated on a thorough evaluation including detailed insights about the experiments in terms of parameters and applied methods.

**Related Work** Finding an object's position in an image is a commonly addressed problem. There are many methods. For example, one may treat the output of an edge detector as an illustration and try to fit the target shape's boundary to the correct position. Such a fitting can even be done using the generalized form of the Hough Transform [4] provided that the shape can be expressed in some parametric form. Such methods, however, are not applicable when the illustration contains embedded shapes (Fig. 1a, b), or when the target figure is a complex form with embedded subfigures. The closest contour matching technique to ours is *chamfer matching* [5]. In chamfer matching, the template is correlated with the distance transform of the illustration. In case of clutter, chamfer matching requires additional improvements, e.g., learning [13]. As one improvement, we propose to replace the distance transform with a more informative diffuse field which implicitly codes curvature. Moreover, our experiments indicate that applying the transformation to the target rather than the illustration may also offer benefits in

certain settings. We provide a complete evaluation. Distance transforms have also been instrumental in level set methods. Most typically, shape knowledge is coded via the signed distance transform, embedding the  $1 - D$  shape boundary as the zero-level set of a function defined on a connected bounded open subset of  $\mathbb{R}^2$  [16, 17]. Level set methods, however, are not applicable when there are embedded shapes.

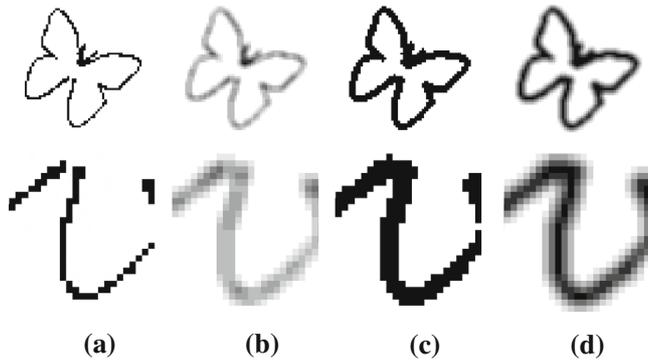
It is possible to replace the sharp interface model in level set based segmentation methods with diffuse ones. For example, smooth distance fields that exhibit exponential decay rather than linear growth are obtained by solving a screened Poisson PDE. These kind of distance fields are more informative in the sense that they implicitly code curvature in addition to distance. The whole topic has a recent revival with a wide range of applications and new theoretical insights [2, 3, 11, 19]. The earliest work by Tari et al. [20] addresses the connection between screened Poisson and image segmentation by the Ambrosio-Tortorelli approximation [1] of the Mumford-Shah model [15]. This particular work has recently been used in [12] to address a search problem where a small fragment of the illustration is searched in order to reveal the underlying global repetition structure in abstract ornaments. The curvature-coding field we propose improves search and does not require solving a PDE.

In reconstructing frescos, Fornasier et al. [9] addressed the problem of locating small fragments within a whole. For each small piece of plaster that still showed an element of the design of the fresco, the authors were able to find where it belonged. This is quite an elegant method, but the non-additive and non-linear nature of the binary illustrations that we consider prevents its use.

To the best of our knowledge, discovery of hidden figures as we describe has not been studied within the mathematical imaging community. Nevertheless, Saarbrücken group's recent inpainting based steganography application [14] addresses the opposite problem: to hide a secret image by embedding it into arbitrary cover images. Both the secret and the cover are dense images, and the recovery of the secret is possible only via a password. That is, ordinary observer cannot detect whether an image contains a secret or not. Object camouflage is also a problem of interest in the computer graphics community [8].

## 2 Formalization

We consider the task of figure hunt games: tracing out *target figures* hidden in binary *illustrations*. Let  $\Omega \subset \mathbb{R}^2$  be the image domain, and let  $\mathcal{F}, \mathcal{I} : \Omega \rightarrow \{0, 1\}$  be the target figure and the illustration, respectively. The goal is to localize a target figure, such as the butterfly Fig. 2a in an illustration, such as the mandala Fig. 4a. Values 1 (white) correspond to the background and 0 (black) to the foreground.



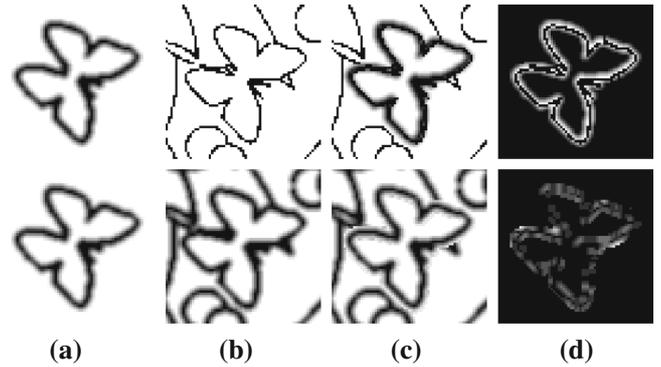
**Fig. 2** Erosion followed by averaging. When pure averaging—e.g. mean filter—is applied to a binary line drawing, the contour location vanishes. In contrast, when erosion with subsequent averaging is applied, the contour location information is retained. **a** Input image. **b** Averaging (mean filter). **c** Erosion. **d** Erosion and averaging

We start by uniformly eroding the white space, or equivalently, dilating the target figure (e.g. Fig. 2a and/or the illustration. Hence, the drawing becomes thicker (see Fig. 2c). Then, we diffuse by computing a local isotropic average. It is sufficient to compute the local average only for the points falling on the thickened figural loci or in a slightly wider band surrounding it. This transforms the sketch-like binary drawing to a gray-tone picture which may be referred as a diffuse drawing  $\mathcal{F}_d$  (Fig. 2d). In the following, we use the term *diffusion* to entitle the transformation of the binary drawing to a gray-tone picture, although this transformation does not necessarily describe a diffusion in the technical sense.

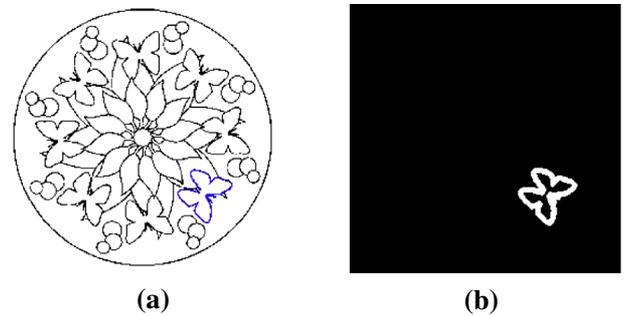
The key idea is to propagate information restricted to figural loci to neighboring areas. Thus, it becomes possible to judge whether a background location is close to or far away from a figural loci. If the averaging and the erosion radii are identical, the highest value is attained on the figural loci; from thereof values decrease as a function of distance in the normal direction. Thus, diffusion produces iso-intensity contours, each following the figural loci from a fixed distance. The lower the intensity, the further away the iso-intensity curve from the figural loci. The second column in Fig. 2 depicts the result of local isotropic averaging applied to the original thin drawing. There, one cannot observe the distance-coding behavior, i.e. the initial thickening is a crucial step.

2.1 Matching Cost

Once the target figure  $\mathcal{F}$  (Fig. 2a) is converted to a diffused form  $\mathcal{F}_d$  (Fig. 2d), the best match is determined by the deformation parameters (i.e. location, orientation and scale) yielding the best *matching cost*. The matching cost is measured as the sum of the gray-value differences between the illustration  $\mathcal{I}$  and the *placed target figure*  $\mathcal{F}_p$  (Fig. 3c). A visualization is shown in Fig. 3. We introduce the matching cost by means of the binary illustration (top row) as well as the diffused illus-



**Fig. 3** Matching cost. Visualized computation of the matching cost. In the *top row*, the diffuse target is searched in a binary illustration. In the *bottom row*, the diffuse target is searched in a diffused illustration. **a** Deformed target figure  $\mathcal{F}_D$ . **b** Zoom of illustration  $\mathcal{I}$ . **c** Placed target figure  $\mathcal{F}_P$ . **d** Matching cost  $E_{cost}$



**Fig. 4** Optimal match. **a** Perfect hint of the target figure in the illustration  $\mathcal{I}$ . **b** Set  $B$  defined by a band surrounding the figural loci

tration (bottom row). A discussion about the role of diffusion follows in Sect. 3.

The placed target figure  $\mathcal{F}_p$  is obtained by the combination of the deformed target figure  $\mathcal{F}_D$  (Fig. 3a) and the illustration  $\mathcal{I}$  (Fig. 3b). Depending on the application,  $\mathcal{F}_p$  can be computed by means of the binary illustration or the diffused illustration. In Matlab coding language  $\mathcal{F}_p$  can be obtained as follows:

$\mathcal{I}$ binary illustration	$\mathcal{I}$ diffused illustration
$\mathcal{F}_p := \mathcal{F}_D;$	$\mathcal{F}_p := \mathcal{I};$
$\mathcal{F}_p(\mathcal{I} \neq 1) = \mathcal{I};$	$\mathcal{F}_p(\mathcal{F}_D \neq 1) = \mathcal{F}_D;$

Hereby,  $\mathcal{F}_D$  is obtained by the deformation of the diffused target figure  $\mathcal{F}_d$  with the respective deformation parameters:

$$\mathcal{F}_D = \mathcal{F}_D(\mathcal{F}_d, \text{deformation parameters}) \tag{1}$$

The visual matching cost in Fig. 3d is defined as the absolute value of the pixel-wise difference between the illustration  $\mathcal{I}$  (Fig. 3b) and the placed target figure  $\mathcal{F}_p$  (Fig. 3c). In general,  $E_{cost}$  can be obtained as follows:

$$E_{\text{cost}} = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} |\mathcal{I}(x) - \mathcal{F}_{\mathcal{P}}(x)|, \tag{2}$$

where  $\mathcal{I} : \Omega \rightarrow [0, 1]$  is the (binary/diffused) illustration,  $\mathcal{F}_{\mathcal{P}} : \Omega \rightarrow [0, 1]$  the placed target figure and  $\mathcal{B} \subset \Omega$  the set indicating the band surrounding the figural loci.

The set  $\mathcal{B}$  is illustrated in white in Fig. 4b.  $\mathcal{B}$  is the collection of pixels which belong to gray values of the deformed target figure (Fig. 3a). Hence, the sum is taken over those locations that fall within the band surrounding the figural loci within which the diffuse field has been constructed. Moreover, the cost is normalized by dividing it by the number of locations that contributed to its computation.

*Example* Let the illustration be the mandala consisting of butterflies shown in Fig. 4a. As target figure we take the butterfly in Fig. 2a. A perfect hint for the position of the butterfly is indicated in blue right on the bottom of the mandala in Fig. 4a. The components of the matching cost (2) for this figure hunt are illustrated in Figs. 4b and 3. The resulting matching cost is visualized in Fig. 3d.

To find a solution with minimal matching cost, we optimize the set of deformation parameters leading to the deformed target figure  $\mathcal{F}_{\mathcal{D}}$ . We determine these optimizing parameters via a probabilistic algorithm which returns multiple solutions. We use genetic algorithm based optimization which is readily available in the Matlab environment. It minimizes an energy functional by varying its input variables. A detailed discussion follows in the next section.

## 2.2 Optimization Via a Genetic Algorithm

A genetic algorithm is a search heuristic that mimics the process of natural evolution. The evolution starts with a population of random generated initial solutions of the problem. In every step new populations are created, such that a fitness function is minimized. The populations are evolved towards an optimal solution by *selection*, *combination* and *modification* of the intermediate results:

- **Selection** identifies good solutions in a population and discards the rest (e.g. by measuring against the fitness function).
- **Combination**—also known as crossover—creates new solutions from existing ones.
- **Modification** (or mutation) introduces new features into the solution to maintain diversity in the population.

This process is repeated as long as either a satisfactory matching cost has been reached, or a maximum number of generations has been produced.

In figure hunt games, the best matching of the target figure with the illustration can be described by the deformation

parameters leading to the best matching cost. Therefore, we aim to solve the following optimization problem:

$$\min_{(\theta, t_r, t_c, h_r, h_c) \in \mathcal{D}} E(\theta, t_r, t_c, h_r, h_c) \tag{3}$$

where  $\mathcal{D} \subset \mathbb{R}^5$  is the domain of the deformation parameters:

- $\theta$  being the rotation angle,
- $t_r, t_c$  describing the translation in row/column direction,
- $h_r, h_c$  for scaling in direction of rows/columns.

The energy  $E$  to be minimized is defined as follows:

$$E(\theta, t_r, t_c, h_r, h_c) := E_{\text{cost}}, \quad (\text{compare Equation (2)})$$

with  $\mathcal{F}_{\mathcal{D}} = \mathcal{F}_{\mathcal{D}}(\mathcal{F}_d, \theta, t_r, t_c, h_r, h_c)$ . (4)

In order to find the optimal set of values in this five dimensional search space, we make use of the genetic algorithm built in Matlab:

```
ga(fitnessfcn, nparams, [], [], [], [],
    lb, ub, [], IntCon)
```

The output of the algorithm includes the parameter set  $(\theta, t_r, t_c, h_r, h_c)$  corresponding to the best matching cost  $E_{\text{cost}}$  for a given figure hunt problem. The input variables have the following meaning:

- fitnessfcn = { @energy\_functional, illustration, target figure }, where the *energy\_functional* is a function which takes the parameter set as well as the illustration and the target figure as input and returns the corresponding matching cost  $E_{\text{cost}}$
- nparams: number of parameters to optimize (= 5)
- lb/ub: lower/upper bound (e.g.  $\theta \in [0, 360]$ )
- IntCon: integer constraints on parameters (= [2, 3]: parameters  $t_r, t_c$  should be integers)<sup>1</sup>

In every step of the genetic algorithm, new populations are created. Hereby, the lower/upper bound constraint as well as the integer constraints have to be fulfilled. The selection, combination and modification process is guided by the values of the *energy\_functional*, aiming to obtain a minimal matching cost  $E_{\text{cost}}$ . We are aware that the genetic algorithm also has disadvantages. In particular, the algorithm is non-deterministic and there is no proof of optimality known. While alternative algorithms are conceivable, we chose the genetic algorithm because it provides a good trade-off between speed and quality of computed solutions.

<sup>1</sup> The integer constraints on  $t_r, t_c$  can also be omitted with the drawback of higher computational costs. However, our experiments showed a sufficient accuracy when restricting the translation to integers.

### 3 The Role of Diffusion

In every step of the genetic algorithm, new populations are created such that the matching cost is minimized. To propagate information restricted to figural loci to neighboring areas we use a diffused representation of the target figure (and the illustration). Thus, it becomes possible to know whether a location is close or far away from one of the desired locations.

Diffusion of the target figure (and the illustration) helps in two different ways:

**1. Uninformative pixels become informative.**

Binary line drawings like the ones shown in Fig. 1 contain large empty (white) regions without any information. By diffusing the drawing, the information restricted to the figural loci becomes visible within a neighborhood. This allows the search for the cloverleaf in Fig. 1a and the hangers in Fig. 1b.

**2. Improved search process.**

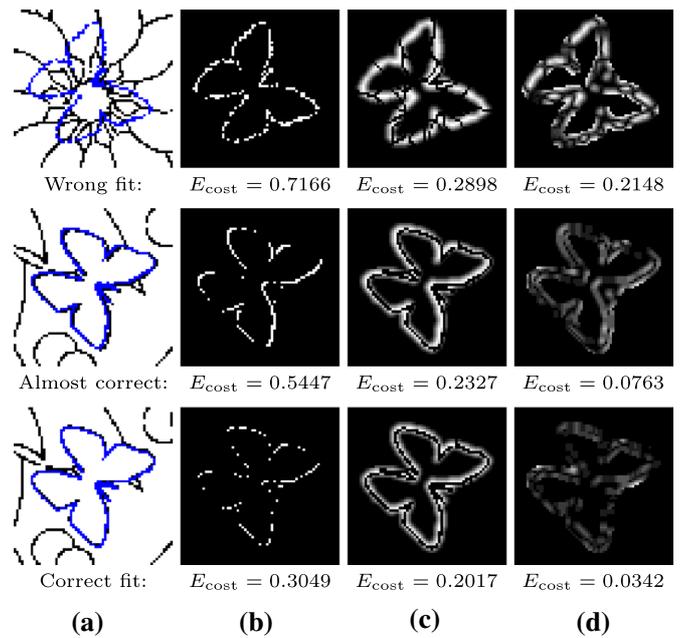
A strong diffusion may simplify and hence speed up the search process (e.g. to get the rough positions of the bees in Fig. 7). In addition, diffusion convexifies the energy and thereby improves the localization.

#### 3.1 Spreading the Edge-Information

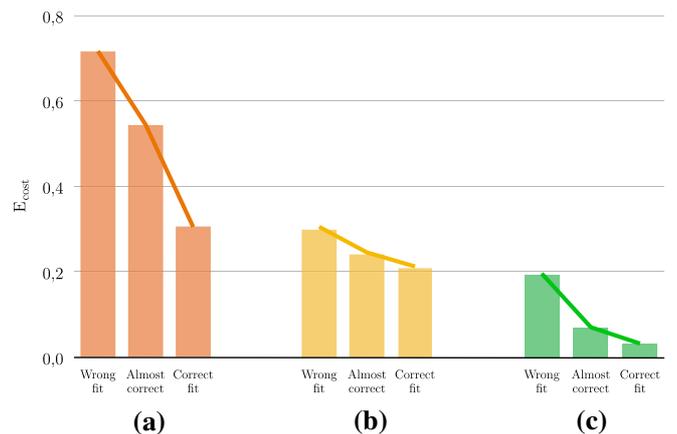
In order to minimize the matching cost, a correlation between the quality of the match and the cost is required. This correlation is not given, if the binary representations of the target figure and the illustration are used.

Figure 5 compares the matching costs of three different matches: (1) wrong fit; (2) almost correct fit; (3) correct fit. The first column indicates the position of the butterfly within the mandala. The second column depicts the matching cost obtained by means of the binary illustration and target figure. Columns three and four give the matching costs obtained with the diffused target figure together with the binary and the diffused illustration. Observe that the matching costs computed with the non-diffused drawings (column b), are almost equal. In particular, the visualized energies of a wrong fit and an almost perfect fit are indiscernible. Hence, there is no reliable optimization criterion. It is unclear whether an intermediate match leads to a good fit. In contrast, the energy of the wrong fit in column c, d is significantly higher (lighter) than the energy of the (almost) correct fit. This means that the cost becomes informative.

Figure 6 demonstrates the different matching costs visualized in Fig. 5. To be able to make decisions about the goodness of a fit, the matching cost corresponding to a bad fit should be substantially higher than the one corresponding to a good fit. In particular, a graph indicating the matching cost of a wrong, an almost correct and a correct fit should first have



**Fig. 5** Expressive energy by diffusion. Observe the energy drop in the last column. **a** Position of  $\mathcal{F}$ . **b** Binary  $\mathcal{F}$ ,  $\mathcal{I}$ . **c** Diffused  $\mathcal{F}$ , binary  $\mathcal{I}$ . **d** Diffused  $\mathcal{F}$ ,  $\mathcal{I}$



**Fig. 6** Comparison of matching costs. The decay of the energy represents the convergence towards a good fit. Compare the first graph to the last. **a** Binary  $\mathcal{F}$ ,  $\mathcal{I}$ . **b** Diffused  $\mathcal{F}$ , binary  $\mathcal{I}$ . **c** Diffused  $\mathcal{F}$ ,  $\mathcal{I}$

a strong decay followed by a weak decay. Figure 6 shows the graphs corresponding to the three columns b–d of Fig. 5. Whereas the two rightmost graphs show the expected decay, the first graph does not imply the position of the target figure. The reason is that the binary figures include lots of uninformative (white) pixels and therefore cannot decide whether a fit is good. In contrast, the diffusion propagates information about the figural loci from purely local to a neighborhood (compare Fig. 2). Hence, the desired location can be observed from some distance and leads to an informative energy.

With the diffused representations, the genetic algorithm has a clear optimization criterion and thus returns the optimal match of the chosen target figure in the given illustration.

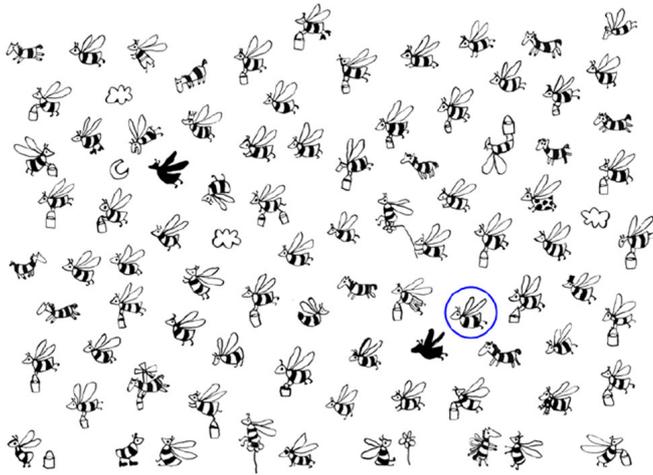


Fig. 7 How many bees are there in the image?

### 3.2 Improved Localization and Speed-Up

A diffused representation not only propagates edge information to a neighborhood, it also simplifies the search process. This behavior comes from the fact that diffusion improves localization by convexifying the energy. Similar blurring strategies are known by continuation approaches such as graduated non-convexity [7].

A typical example is the swarm of bees in Fig. 7. To count the number of bees, a strong diffusion can be applied, leading to an accumulation of gray splotches. Taking one of them as target figure, the genetic algorithm quickly detects the splotches throughout the bee swarm. In a second step, a finer search can be applied around the detected positions to obtain a more precise hint of the bees. Extensive experiments will be shown in Sect. 5.1.1. A crucial point is the *preservation of the original contour* features. Despite diffusing the binary drawings the contour location has to be preserved. This is not given for all diffusion methods. Edges can be washed-out without coding the original contour location or discretization artifacts can be amplified.

To choose the best diffusion method for the particular considered task, we will discuss different diffusion methods in the next section.

## 4 Diffusion Methods

We discussed that diffusion is an essential step for the algorithmic solution of a figure hunt game. During our studies we tested several diffusion approaches. In the course of this section we will give a detailed discussion of the four most interesting ones:

- Averaging
- Distance function

- v-transform [20]
- Erosion followed by averaging

In the following, let  $\Omega \subset \mathbb{R}^2$  denote the image domain and  $I : \Omega \rightarrow \{0, 1\}$  be the binary image.

### 4.1 Averaging

In order to spread the edge-information, the most intuitive way is to apply a diffusion filter, e.g., the mean filter, where each pixel value is replaced by the mean of the pixel values in its neighborhood. Let  $\sigma > 0$  be a parameter. The averaging is a basic convolution of  $I$ , where each pixel  $(x, y) \in \Omega$  is assigned the average value of its neighborhood of size  $\sigma \times \sigma$ . Hence, the diffused version can be obtained by:

$$(I * k)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x - a, y - b) k(a, b) da db, \quad (5)$$

where  $(x, y) \in \Omega$  and

$$k(a, b) = \begin{cases} \frac{1}{\sigma^2} & \text{if } |a| \leq \frac{\sigma}{2} \text{ and } |b| \leq \frac{\sigma}{2}, \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

is a standard kernel of the mean filter. For the boundary condition, *zero padding* is used, i.e. the boundary of the image is augmented by zeros.

The butterfly diffused by averaging is shown in Fig. 2b. The parameter  $\sigma$  was set to  $\sigma = 3$ . In the zoomed part of the figure (second row), one cannot identify the original location of the contour. The contour is completely blurred, and it merges with the background. One would rather like to spread the edge information but at the same time keep the information about the original edge location. Hence, we propose a second method: *erosion followed by averaging*, which will be described in Sect. 4.4.

### 4.2 Distance Function

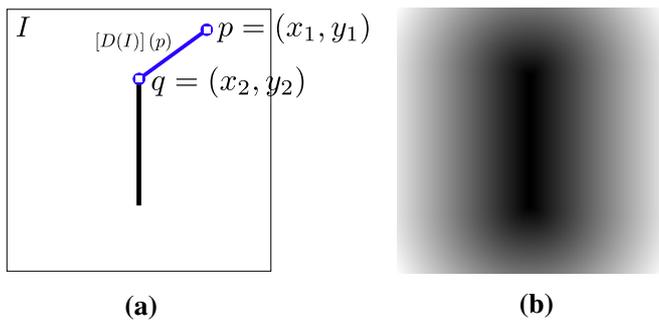
Another option for spreading the edge information is the usage of the distance function (sometimes referred to as the *distance transform*). The *distance function*  $D$  of a binary image  $I$  associates each pixel  $p$  of the domain  $\Omega$  of  $I$  with its distance to the nearest zero-valued pixel:

$$[D(I)](p) = \min \{d(p, q) \mid I(q) = 0\}. \quad (7)$$

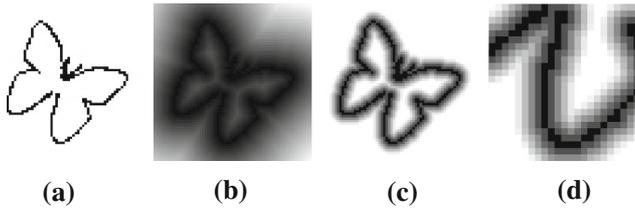
The distance function of an image of a black line on a white background is illustrated in Fig. 8.

Here, the metric  $d$  for a space  $\mathbb{E}$  is a function associating a nonnegative real number with any two points  $p$  and  $q$  of  $\mathbb{E}$  and satisfying the three conditions of a norm. E.g. the *Euclidean distance*  $d$ :

$$d[(x_1, y_1), (x_2, y_2)] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (8)$$



**Fig. 8** Exemplary distance transform. **a** Input image  $I$ . **b** Distance transform of  $I$



**Fig. 9** Distance transform of the butterfly line drawing. By restricting the values to a band surrounding the figural loci, the desired diffused target figure results. However, the zoom shows a strong effect of discretization noise. **a** Input image. **b** Distance transform. **c** Masked distance transform. **d** Zoom of **c**

where  $p = (x_1, y_1)$  and  $q = (x_2, y_2)$ .

The distance transform of the butterfly is shown in Fig. 9b. Each pixel includes information about its distance to the contour. The desired diffused version, however, requires the information to be restricted to a band around the contour. Thus, we omit the values beyond a band surrounding the figural loci and stretch the remaining gray-values to fill the whole range from 0 to 255. The resulting 'masked' version is shown in Fig. 9c with a closeup in (d). The closeup reveals the effect of discretization noise which remained (even amplified) despite the diffusion.

4.3  $v$ -Transform [20]

To obtain a gray-tone picture of a sketch-like binary drawing, one other option is to use the  $v$ -transform. It is the minimizer of the following functional

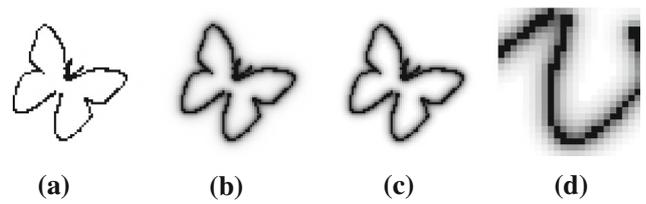
$$\frac{1}{2} \int \int_{\Omega} \left\{ \rho \|\nabla v\|^2 + \frac{(1-v)^2}{\rho} \right\} dx dy \tag{9}$$

subject to  $v(x, y) = 0$  on  $\{(x, y) : I(x, y) = 0\}$ .

Numerically, we solve for  $v$  iteratively using the following update step:

$$v^{t+1} = \left( 1 + \frac{\tau}{\rho^2} \right)^{-1} \cdot \left( v^t + \frac{\tau}{\rho^2} + \tau \nabla^2 v^t \right) \tag{10}$$

where  $\tau$  denotes the step size.



**Fig. 10**  $v$ -transform of the sketch-like binary butterfly. **a** Input image. **b**  $v$ -transform. **c** Masked **b**. **d** Zoom of **c**

For the butterfly drawing, the resulting gray-tone picture computed with the parameters  $\tau = 0.5$  and  $\rho = 3$  is shown in Fig. 10b. Again, diffused pixels are spread throughout the whole image. Hence, we mask the image and omit values too far away. The result is displayed in Fig. 10c with a closeup in d.

Unlike the usual distance transform (Sect. 4.2),  $v$  is an implicit coder of the curvature, a valuable geometric feature, without explicit estimation of higher order derivatives. (One of the original goals in proposing  $v$  was to bridge low level and high level vision [18,20]).

In this paper, in the setting of our specific task, we advocate a much simplistic way of obtaining an analogous behavior in a band around the contour of the drawing. We present this idea in the next section. Our computation does not require the computation of the entire  $v$  function on the entire domain  $\Omega$  by solving a PDE.

4.4 Erosion Followed by Averaging

To keep the edge information while blurring the contour, we combine pure averaging presented in Sect. 4.1 with the morphological operation 'erosion'. The intuitive idea is to: (1) broaden the edge of the binary line drawing; (2) smooth the thicker edge. If the averaging and the erosion radii are identical, the highest value is attained on the figural loci; from thereof values decrease as a function of distance in the normal direction. Fig. 2c, d illustrate the broadened edge and the diffused version for the butterfly drawing.

In the first step, an erosion is applied to broaden the contour line of the binary line drawing. In principle, one is used to the term 'dilation' for enlarging. To stick with the standard terminology referring to the white space, we use the term 'erosion' (instead of 'dilation' of the black space). In the following, we show a detailed explanation of the applied erosion.

Let  $S$  be the structuring element. We denote the erosion of the image  $I$  by  $S$  via  $\varepsilon_S(I)$ :

$$\varepsilon_S(I) = \bigwedge_{s \in S} I_{-s}, \tag{11}$$

the minimum of the translations of  $I$  by the vectors  $-s$  of  $S$ . In other words, the eroded value at a given pixel  $x$  is the

minimum value of the image in the window defined by the structuring element  $S$  when its origin is at  $x$ :

$$[\varepsilon_S(I)](x) = \min_{s \in S} I(x + s). \tag{12}$$

The butterfly eroded by the set

$$S = \{(x, y) \in \mathbb{Z}^2 \mid \|(x, y)\| \leq S\} \tag{13}$$

with  $S = 1$  is shown in Fig. 2c.

In the next step, the eroded image is smoothed (as in Sect. 4.1):

$$(\varepsilon_S(I) * k)(x) = \int_{\mathbb{R}^2} [\varepsilon_S(I)](x - a) k(a) da, \tag{14}$$

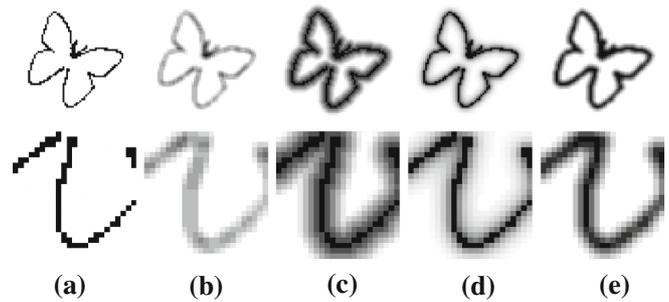
where  $x \in \Omega$  and  $k$  is a standard kernel of the mean filter as defined in Eq. (6) with  $\sigma = 2S + 1$ . Again, zero padding is used to augment the boundary. The butterfly obtained by erosion with  $S = 1$  (in Eq. (13)) and subsequent averaging with  $\sigma = 3$  is shown in Fig. 2d.

#### 4.4.1 Advantages

Within a band surrounding the figural loci, our diffuse drawing (obtained by erosion of the white space followed by averaging) mimics a curvature coding distance field similar to the  $v$ -transform, the solution of a screened Poisson PDE [20]. We avoid solving Poisson PDEs or variants for two reasons. Firstly, our approximation is both easier and faster to compute. But more importantly, a Poisson based distance field, being the steady state solution to a biased diffusion equation is too much influenced by long-range interactions among opposing boundaries. This may be detrimental if several figural loci overlap as in Fig. 1 top row.

#### 4.5 Overview

In Fig. 11, we compare our diffuse drawing to the alternatives: pure averaging, usual distance image and the  $v$ -transform. All diffuse drawings are restricted to a band surrounding the figural loci. Whereas the pure averaging (b) returns a blurred image where the contour location vanished, the other diffusion approaches keep the edge information while blurring. However, the effects of discretization noise remain (even amplified) in the usual distance image (c), whereas the iso-intensity contours in our diffuse model (e) smoothly follow the boundary. Additionally, our diffuse model (e) is obtained by a simplistic approach. In contrast, the approach (d) requires solving a PDE.



**Fig. 11** Comparison of introduced diffusion approaches. The proposed erosion followed by averaging **e** gives the most informative diffused image whilst being much simpler than **c** or **d**. **a** Input image. **b** Pure averaging. **c** Distance transform. **d**  $v$ -transform. **e** Erosion and averaging

### 5 Experimental Results

In this section, we show extensive validations and demonstrate the performance of the proposed concept for solving figure hunt games.

The parameters used for the experiments are summarized in Table 1. Unless specified otherwise, diffusion is computed by *erosion followed by averaging* with Eqs. (12)–(14) where  $S = 1$  (and  $\sigma = 3$ ).

#### 5.1 Propagation of the Contour Information

Depending on the application, one has to specify (a) the intensity of the diffusion and (b) if the placed target figure  $\mathcal{F}_P$  should be computed by means of the binary or the diffuse illustration. Therefore, we categorize the figure hunt games as follows:

- *Where’s Wally?* type images (Fig. 1 bottom row)
- Gottschaldt type puzzles (Fig. 1 top row)

##### 5.1.1 Where’s Wally? Type Images

*Where’s Wally?* type illustrations consist of several objects being placed next to each other like Figs. 7 and 15. Different questions can arise here, like e.g.: “How many bees are in the bee swarm?”, “Can you find the objects not belonging to the scene?” or “How many cars of the same type are there?”. These questions can be allocated to two general problem settings:

1. Get a rough idea about the drawing (Fig. 33d).
2. Find the exact position of a given target figure (Fig. 33c).

Both problems can be approached in a first step by localizing the approximate positions of the objects using a strong

**Table 1** Parameters of the optimization function and the average runtime per experiment. The diffusion is computed by means of Eq. (14)

Results		Input				Diff. Prop.		Param. Ranges					
Fig.	Time	Illustration $\mathcal{I}$	Fig.	$ \mathcal{I}  = \mathcal{I}_c \times \mathcal{I}_r$	Targ. fig. $\mathcal{F}$	Fig.	of $\mathcal{F}$	of $\mathcal{I}$	$\theta$	$t_r$	$t_c$	$h_r$	$h_c$
12	24s	Multiple bees	7	$766 \times 556$	Bee	7	$S = 5$		$[0, 360]$	$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	$[0.7, 1.5]$	$h_r$
16	21s	Collection of Cars	15a)	$483 \times 254$	Car	15b)	$S = 4$			$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	$[0.7, 1.3]$	$h_r$
15a)	64s	Collection of Cars	15a)	$483 \times 254$	Car	15b)	$S = 1$			Eq. (15)	Eq. (16)	$[0.7, 1.3]$	$h_r$
23a)	38s	Hidden cloverleaf	18 l.	$188 \times 188$	Cloverleaf	18 r.	$S = 1$	-		$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	1	1
31	45s	Hidden pi	18 l.	$188 \times 188$	Pi ( $\pi$ )	30a)	$S = 1$	-				$[0.7, 1.3]$	$h_r$
23b)	205s	Several hangers	21	$304 \times 321$	Hanger	22a)	$S = 1$	-				$[0.8, 1.2]$	$[0.8, 1.2]$
25	176s	Butterfly mandala	24	$250 \times 250$	Butterfly	24 r.	$S = 1$					1	1
32	210s	Butterfly mandala	24	$250 \times 250$	Segment	24 l.	$S = 1$					1	1
33c)	97s	Hidden elephant	1c)	$273 \times 205$	Elephant	33c)	$S = 1$		$[0, 360]$			1	1
27	209s	Mandala circles	26a)	$550 \times 550$	Circle	26b)	$S = 1$		0			$[0.13, 2.9]$	$h_r$
28	811s	Triangles	26a)	$1706 \times 640$	Triangle	26b)	$S = 4$	-	$[0, 360]$	$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	$[0.3, 1.1]$	$[0.3, 3.2]$

diffusion (see e.g. Figs. 12 and 16). If additionally, the exact locations of the objects are desired, the resulting approximate positions can be used to constrain the search space for the search on the fine scale.

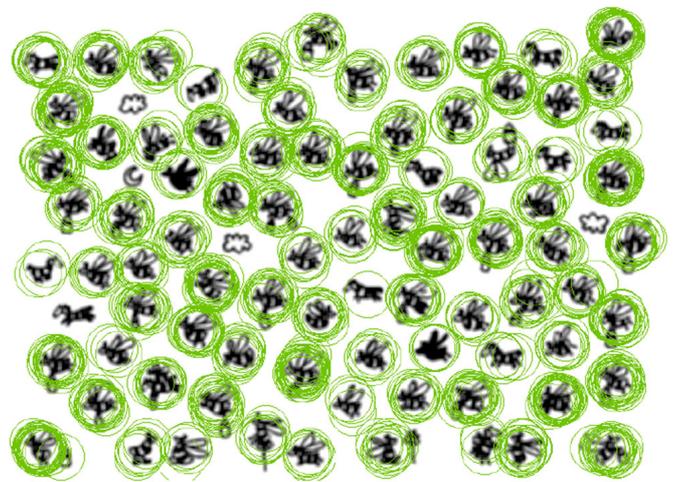
*Swarm of Bees* In order to analyze the swarm of bees, we searched for a strongly diffused version of the target figure marked by the blue circle in Fig. 7 in the diffused illustration shown in the background of Fig. 12. For the diffusion we use Eq. (14) and set  $S = 5$ . The ranges of the parameters ( $\theta, t_r, t_c, h_r, h_c$ ) are set as follows:  $\theta \in [0, 360]$ ,  $t_r \in [-\mathcal{I}_r, \mathcal{I}_r]$ ,  $t_c \in [-\mathcal{I}_c, \mathcal{I}_c]$ ,  $h_r \in [0.7, 1.5]$  and  $h_c := h_r$ .

Due to the strong diffusion, we are able to quickly detect the rough locations of the bees—marked by green circles in Fig. 12—in this large illustration ( $766 \times 556$  pixels). The number of circles around each bee is an evidence that good fits are found more often than bad ones. In the next step, we omit the duplicates and mark each location by exactly one circle. We obtain 84 distinct circles corresponding to 84 objects in the bee swarm.

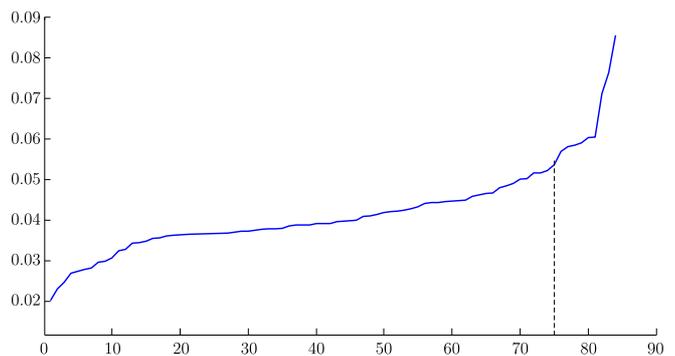
By analyzing the matching costs corresponding to the 84 hypotheses we obtain the plot in Fig. 13. For the first 74 hypotheses the energy increases steadily. In contrast, the energy ascends steeply for the last 10 hypotheses. Hence, we declare the worst 10 fits as objects not belonging to the bee swarm and indicate their position in orange. Figure 14 shows the 84 determined locations, whereof the worst 10 fits are indicated in orange. Observe that all orange marked objects are no bees.

In a second step, the approximate locations could be used to obtain a more exact location and orientation of the bees. Such a search on a fine scale will be explained in the next paragraph by means of the *collection of cars*.

*Collection of Cars* To find all cars of the same type as the target car (Fig. 15b) we use a strong diffusion for the target figure and the illustration, which is exemplary shown in Fig. 15c. Therefore, we use Eq. (14) and set  $S = 4$ . On the

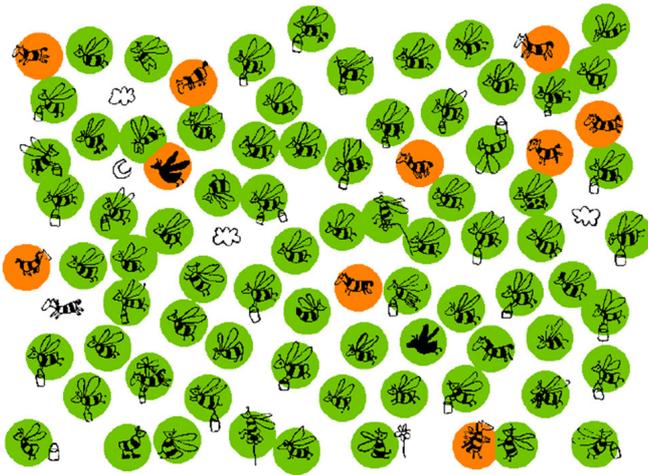


**Fig. 12** All hypotheses of 1,000 independent runs of the algorithm on a coarse scale

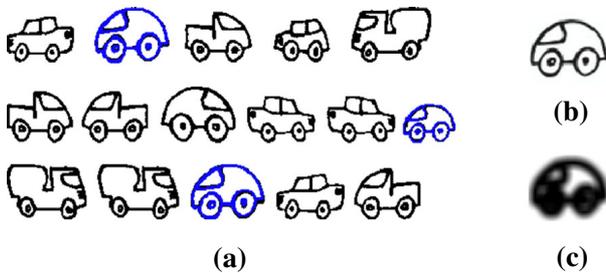


**Fig. 13** Matching cost corresponding to the 84 distinct locations. The matching cost is plotted against the hypotheses, ordered by increasing cost

coarse scale, the algorithm returns the locations illustrated in Fig. 16. In a second step we use these locations to initialize the algorithm for the search on a fine scale. For each position obtained by the search on the coarse scale, an additional search with a slightly diffused target figure and illustration is



**Fig. 14** Several dissimilar objects found. The number of bees in the bee swarm can easily be counted by using our search algorithm together with a strong diffusion. We detected 84 objects, including 10 objects which do not belong to the bee swarm



**Fig. 15** Three cars of the same type are detected with the search on the fine scale. The search was initialized by the positions obtained by the search on the coarse scale (Fig. 16). Additionally, the search space was constrained to a small area around these positions. **a** Illustration with best three results. **b** Target figure. **c** Coarse scale

carried out ( $S = 1$ ). Thereby, we constrain the search space to a small area around these positions.

Let  $(T_r, T_c)$  be a position obtained by the search on the coarse scale and let  $|\mathcal{F}| = \mathcal{F}_c \times \mathcal{F}_r$  be the size of the target figure  $\mathcal{F}$ . For the search on the fine scale, we restrict the parameter ranges of  $t_r$  and  $t_c$  as follows:

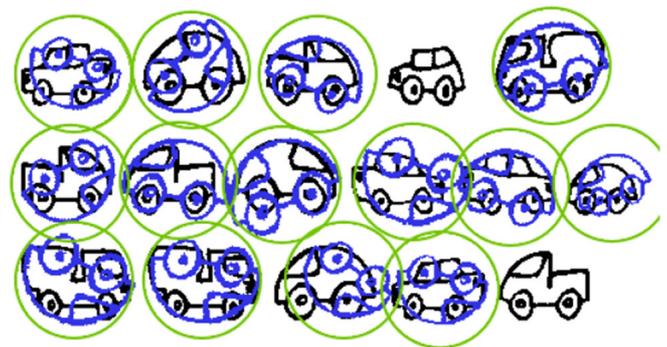
$$t_r = \left[ T_r - \left\lceil \frac{\mathcal{F}_r}{4} \right\rceil, T_r + \left\lceil \frac{\mathcal{F}_r}{4} \right\rceil \right], \tag{15}$$

$$t_c = \left[ T_c - \left\lceil \frac{\mathcal{F}_c}{4} \right\rceil, T_c + \left\lceil \frac{\mathcal{F}_c}{4} \right\rceil \right]. \tag{16}$$

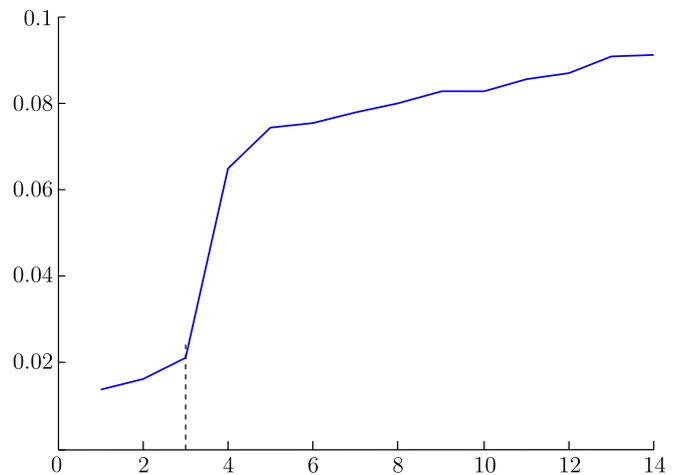
The matching costs corresponding to the resulting hypotheses are plotted in Fig. 17. A distinctive jump of the cost can be observed at the fourth hypothesis.

The three hypotheses with the best energy are thresholded and the binary shapes are depicted in blue in Fig. 15a. Three cars of the given type occur in the image.

This two-step approach allows to get a rough analysis of the illustration followed by a precise definition of the deformation parameters in the second step. For the search on the



**Fig. 16** Coarse-to-fine approach. The search for a single target in a collection of multiple individual objects is performed in two steps: Search on (1) coarse scale; (2) fine scale. The search on the coarse scale returns the approximate positions of objects similar to the target figure



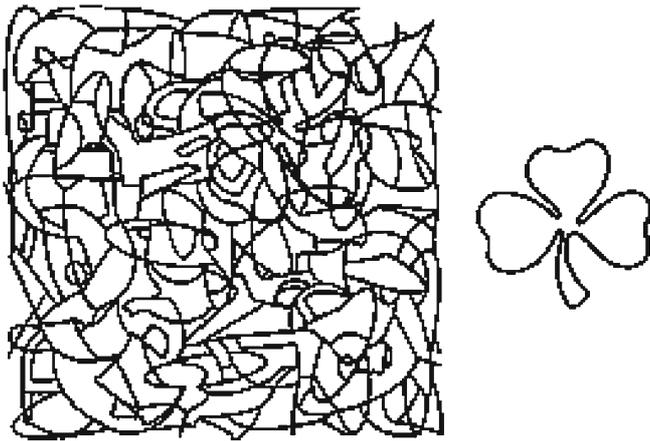
**Fig. 17** Distinctive jump in the energy. The graph of the matching costs corresponding to the hypotheses resulting from the search on the fine scale has a distinctive jump at the fourth hypothesis

coarse scale, a strong diffusion of both the illustration and the target figure is helpful to have as much information as possible throughout the image. For the subsequent search on the fine scale slightly diffused versions are used. The parameters used for both steps are summarized in Table 1.

### 5.1.2 Gottschaldt Type Puzzles

Gottschaldt type puzzles are line drawings where several lines overlap, like in Figs. 18 and 21. A figure hunt game might for example challenge to find the cloverleaf hidden in the line drawing in Fig. 18. Another task might be to find one of the target figures drawn in Fig. 22 in the collection of hangers, Fig. 21. Two different problem settings can appear:

1. The target figure  $\mathcal{F}$  is a cutout of the illustration  $\mathcal{I}$  (compare Fig. 22c). *I.e.* it reflects a complete segment of the illustration.



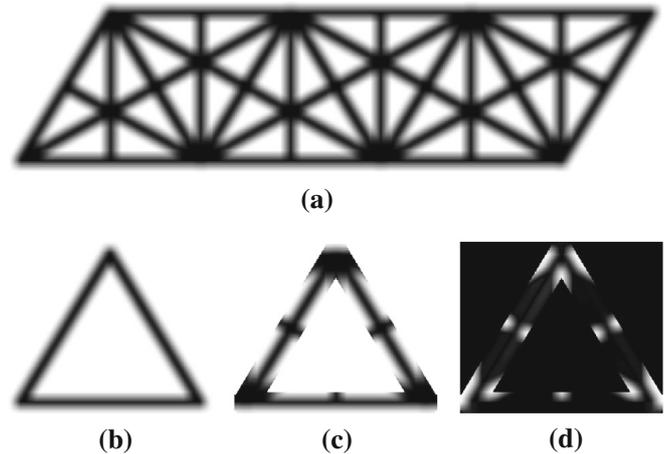
**Fig. 18** Hidden cloverleaf. Can you trace out the hidden cloverleaf?

- The target figure  $\mathcal{F}$  does not reflect a complete segment of the illustration  $\mathcal{I}$  (e.g. Fig. 22a. I.e. at a perfect position of  $\mathcal{F}$  in the illustration,  $\mathcal{I}$  has additional crossing lines (compare Fig. 23).

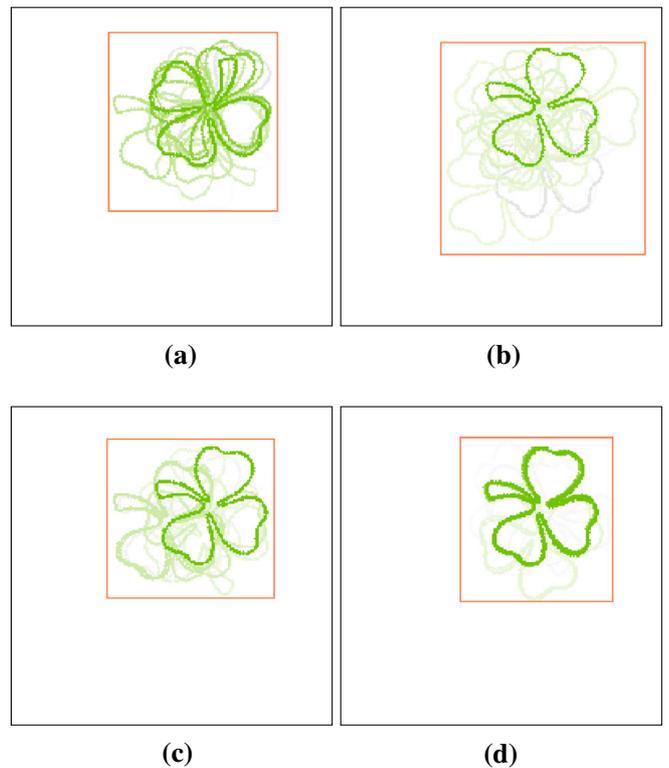
The first type of problems where  $\mathcal{F}$  is a cutout of  $\mathcal{I}$  is mentioned in Sect. 5.3 and an example is shown in Fig. 32.

For the second type of problems, a diffusion of both, the target figure and the illustration may lead to unwanted effects: Due to several overlaps of the lines within the drawing, dark black patches appear at the intersections. However, the diffused target figure does not reflect a complete segment of the illustration and hence does not have such black patches along the contour line. See Fig. 19 for an exemplary illustration. Hence, the matching cost of a perfect fit in a Gottschaldt type puzzle is considerably higher than the matching cost of a perfect fit in a *Where's Wally?* type image. For these cases, we recommend the computation of the matching cost by means of the binary illustration (compare Fig. 3 top row). By only diffusing the target figure, information restricted to figural loci can be propagated to neighboring areas and at the same time the black-spot-problem can be preserved.

In Fig. 20 and Table 2 we demonstrate the results of the different combinations of a binary and a diffused target figure and illustration by means of the cloverleaf line drawing (Fig. 18). Strong green colors indicate a position which leads to a lower energy compared to the other hypotheses. The orange box indicates the *overall size of the spread*, i.e. the area where hypotheses are placed. The area spread percental to the size of the illustration  $\mathcal{I}$  and the average deviation from the optimal position are summarized in Table 2. All hypotheses obtained with the binary target figure together with the binary illustration (a) are misplaced and have approximately the same matching cost. In contrast, for the remaining cases, the hypotheses belonging to the correct position have a significantly lower energy than the misplaced hypotheses. Due



**Fig. 19** Black spots at intersections. The diffused illustration shows black patches at the intersection of the lines. However, the diffused target figure does not include those dark spots. Hence, the visual matching cost  $\mathbf{d}$  of a perfect fit includes white spots. **a** Diffused illustration. **b** Diffused target figure. **c** Cutout of illustration. **d** Difference  $\mathbf{b-c}$

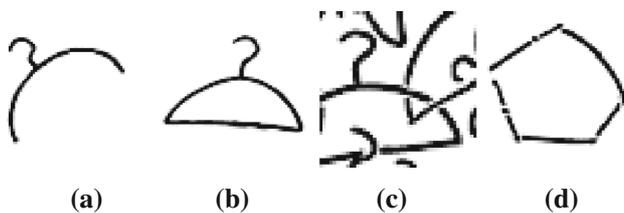


**Fig. 20** Diffusion of target figure and/or illustration computed by means of Eq. (14) with  $S = 1$ . Hypotheses resulting from 10 individual runs of the algorithm. Strong green color indicates low matching cost. Observe that the best hypotheses are obtained by only diffusing the target figure. **a** Binary target figure and illustration. **b** Diffused target figure and illustration. **c** Binary target figure, diffused illustration. **d** Diffused target figure, binary illustration (Color figure online)

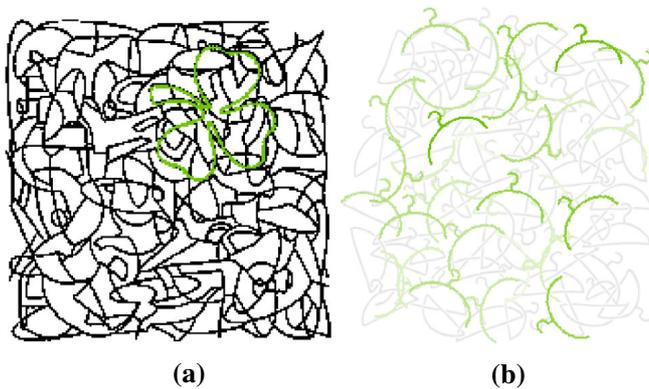
to the diffusion in (b–d) the contour information is propagated to the neighborhood making uninformative (white) pixels informative. Due to the black-spot-problem, the output



**Fig. 21** Collection of hangers



**Fig. 22** Possible target figures to search for: two types of hangers, a segment or a pentagon. **a, b** Different hangers. **c** Segment. **d** Pentagon



**Fig. 23** Hypotheses obtained by spreading the edge information via diffusion. In **b** strong green colors indicate a position which leads to a lower energy compared to the other hypotheses. **a** Best hypothesis. **b** 32 best hypotheses (Color figure online)

in (b) has the largest average deviation and area spread. The best matches are obtained in (d) for the usage of the diffused target figure together with the binary illustration (Fig. 21).

*Hidden Cloverleaf* To find the hidden cloverleaf, we use the approach in Fig. 20d and compute the matching cost by means of the binary illustration (Fig. 3 top row). The best hypothesis is depicted in Fig. 23a. The average runtime is 38.3 seconds.

*Collection of Hangers* The same combination of the diffused target figure and the binary illustration is used for the computation of the matching cost in the search for the hangers. The search for the hanger leftmost in Fig. 22a leads to the hypotheses shown in Fig. 23b. Strong green colors indicate a position which leads to a lower energy compared to the other hypotheses.

### 5.2 Proof of Concept

In this section, we will demonstrate that our approach can handle the similarity transformations translation, rotation and scaling. Furthermore, based on our results we will show that the genetic algorithm together with the defined cost is reliable for the particular problem.

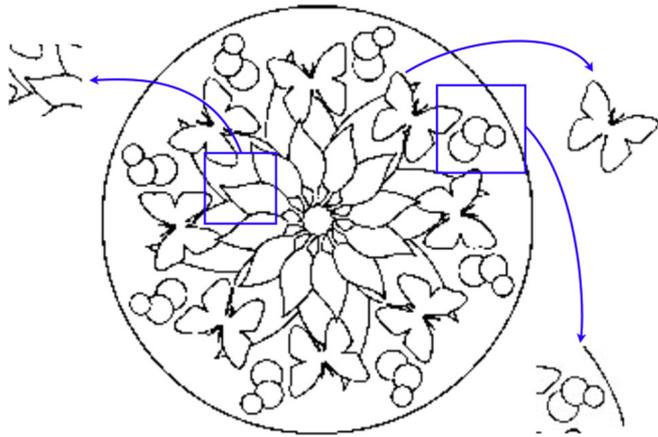
#### 5.2.1 Robustness to Pose Variations

To observe the robustness with respect to pose, we consider a simple mandala pattern (Fig. 24). Possible target figures are indicated by the blue boxes and arrows. For the first experiment the butterfly is chosen as target figure. We diffuse the target figure as well as the illustration by erosion followed by averaging (Sect. 4.4) with  $S = 1$ . Figure 25 shows the hypotheses of 100 individual runs of the genetic algorithm described in Sect. 2.2. To enhance the visibility, the hypotheses are thresholded and the binary shapes are drawn in shades of green. The different shades of green show the energy weighted by the number of detections. Strong green colors indicate: (a) a position which leads to a lower energy compared to the other hypotheses; and (b) a match which has been returned more often in comparison to the other ones. Independent of the position and orientation, all but-

**Table 2** Average results by means of the binary/diffused target figure and illustration. Average deviation of the placed target figure from the optimal position, area spread percental to the size of  $\mathcal{I}$  and average runtime per run

	Average deviation	Area spread perc. to $ \mathcal{I} $	Average runtime
(a) Binary $\mathcal{F}, \mathcal{I}$	12.464	0.327	38.428
(b) Diffused $\mathcal{F}, \mathcal{I}$	23.588	0.471	38.395
(c) Binary $\mathcal{F}$ , diffused $\mathcal{I}$	14.809	0.289	39.961
(d) Diffused $\mathcal{F}$ , binary $\mathcal{I}$	<b>7.344</b>	<b>0.271</b>	<b>38.297</b>

The best results are given in bold



**Fig. 24** Several target figures taken from a mandala consisting of butterflies



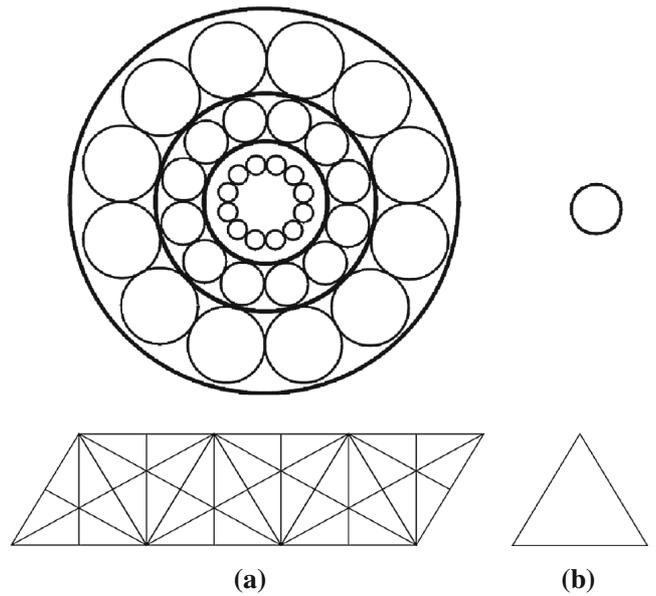
**Fig. 25** Robustness to pose variations. Independent of their location and rotation, all butterflies are found. Correct fits appear more often and have a lower energy

terflies in the mandala are successfully detected. Figure 25 provides experimental evidence that our method is robust to pose variations in the translational and rotational sense.

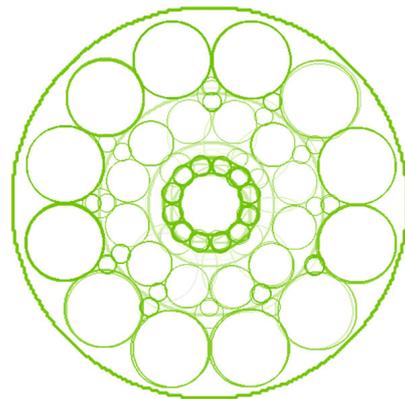
### 5.2.2 Robustness to Scale Variations

To evaluate the robustness to scaling we consider a composition of circles/triangles of varying size. One of the circles/triangles is selected as the target figure, see Fig. 26. The goal is to find all occurrences irrespective of their scaling.

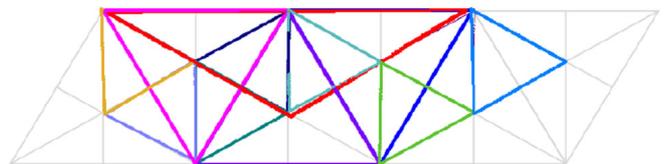
In Fig. 27, we depict the energetically best 99 percent of circles detected after 1,200 runs of the genetic algorithm. The same color coding as in Fig. 25 was used. Observe that the method can handle scale variations. In particular, an unexpected and a very interesting solution is obtained: the innermost circle defined by the twelve smallest circles. This emergent circle may not be immediately perceivable.



**Fig. 26** Illustrations including figures of different scales. The goal is to find all circles/triangles of arbitrary scale. **a** Illustration. **b** Target figure



**Fig. 27** Robustness to scale variations. *Circles* of various scales are detected



**Fig. 28** Robustness to scale variations. *Triangles* of arbitrary scale are detected. Due to clarity, only part of the best hypotheses are depicted in different colors (Color figure online)

In Fig. 28, we depict selected triangles obtained by several runs of the algorithm. Observe that triangles with diverse edge length have been detected, e.g. the red triangle was elongated twice in  $x$ -direction and contracted in  $y$ -direction. The parameters used for these experiments together with the obtained average runtimes are summarized in Table 1.

### 5.2.3 Tendency to Return Good Fits

We evaluated whether the good fits (those of lower matching cost) are obtained more often than the bad fits. This is important as the algorithm is not a deterministic one.

We performed independent runs of the genetic algorithm, each run producing several hypotheses. We then computed the average of the batches of independent runs. As the results shown in Figs. 25 and 27 compellingly demonstrate, the algorithm has a tendency to return good fits more often than the bad ones. Furthermore, none of the bad fits has a nice green color. Hence, we can conclude:

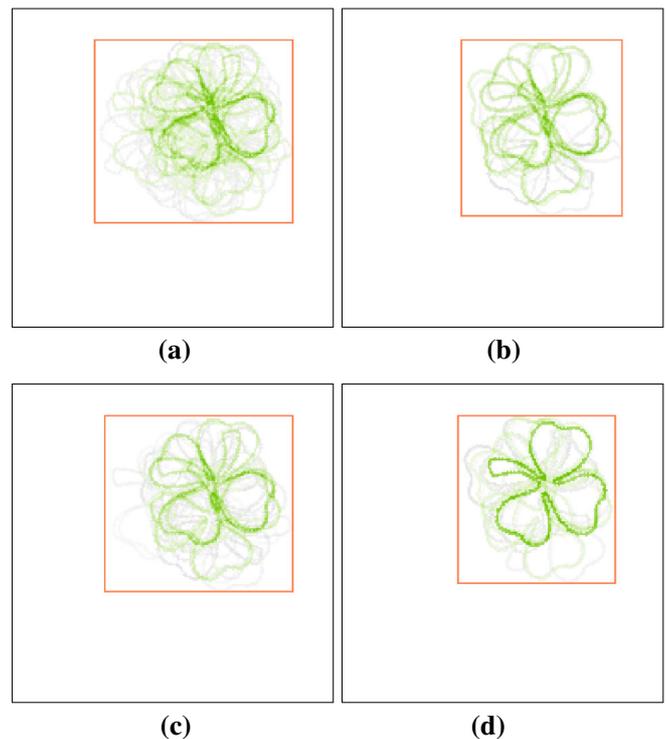
1. The genetic algorithm has a tendency to return good fits:
  - (a) Good fits appear more often than bad fits.
  - (b) The same wrong fit is not detected several times.
2. The matching cost is an indicator for the goodness of the match:
  - (a) A low matching cost indicates a good fit.
  - (b) A large matching cost indicates a bad fit.

### 5.2.4 The Most Descriptive Diffusion Approach

In Sect. 3 we discussed the role of diffusion as key-ingredient of algorithmic search for target figures in a drawing. In this section we will compare the experimental results obtained with the introduced diffusion methods by means of the cloverleaf line drawing (Fig. 18). For the diffusions the parameters are set as given in Sect. 3.

The hypotheses of 60 individual runs of the genetic algorithm obtained by using the different diffusion approaches introduced in Sect. 3 are depicted in Fig. 29. Strong green colors indicate a match which has been returned more often in comparison to the other ones. In contrast to the figures shown above, the color-coding does not include the energy of the single hypotheses. The orange box again indicates the overall size of the spread. The area spread percental to the size of the illustration  $\mathcal{I}$  and the average deviation from the optimal position are summarized in Table 3.

The hypotheses in Fig. 29a–c have all about the same color, i.e., none of the hypotheses was found more often than the others. In contrast, the hypotheses obtained with our proposed diffusion approach (d) accumulate at the correct position. This fact reflects in the average deviation and the area spread indicated by the orange boxes and summarized in Table 3. Compared to the other diffusion approaches, the proposed *erosion and averaging* leads to a significantly smaller average deviation from the optimal fit and to the smallest area spread. The average runtimes are about the same for the different diffusion approaches, however, using the v-transform



**Fig. 29** Results of 60 runs with different diffusion approaches. Strong green colors indicate a match which has been returned more often in comparison to the other ones. The color-coding does not include the energy of the single hypotheses. Hence, with the proposed diffusion approach *erosion and averaging*, it is more likely that the target figure is placed at the correct position. **a** No diffusion. **b** Distance transform. **c** v-transform. **d** Erosion and averaging (Color figure online)

**Table 3** Smallest deviation with erosion and averaging. Average results of 60 runs with the different diffusion approaches shown in Fig. 29

	Average deviation	Area spread perc. to $ \mathcal{I} $
(a) No diffusion	16.481	0.354
(b) Distance transform	12.432	0.277
(c) v-transform ( $\tau = 0.5, \rho = 4$ )	12.732	0.324
(d) Erosion and averaging	<b>7.632</b>	<b>0.258</b>

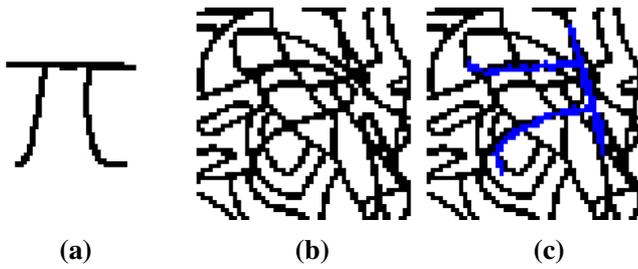
The best results are given in bold

the runtime increases due to the required solution of the PDE by a factor of twenty.

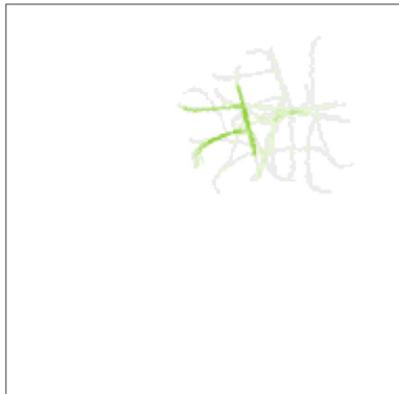
The resulting numbers point to the fact that our proposed diffusion approach *erosion and averaging* gives the best results.

### 5.3 Diverse Target Figures

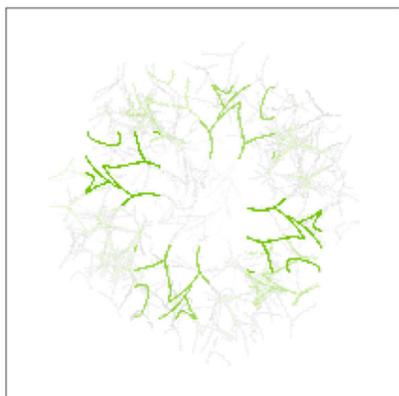
Up to now, we focused on well-known shapes being included in the illustration. Of course our algorithm can also handle segments cut out of the illustration or target figures which are actually not contained in the drawing.



**Fig. 30** Closest match. A very good match is found. Indeed, the letter 'pi' is hidden in the line drawing. **a** Target figure. **b** Zoom of the drawing. **c** Located target figure



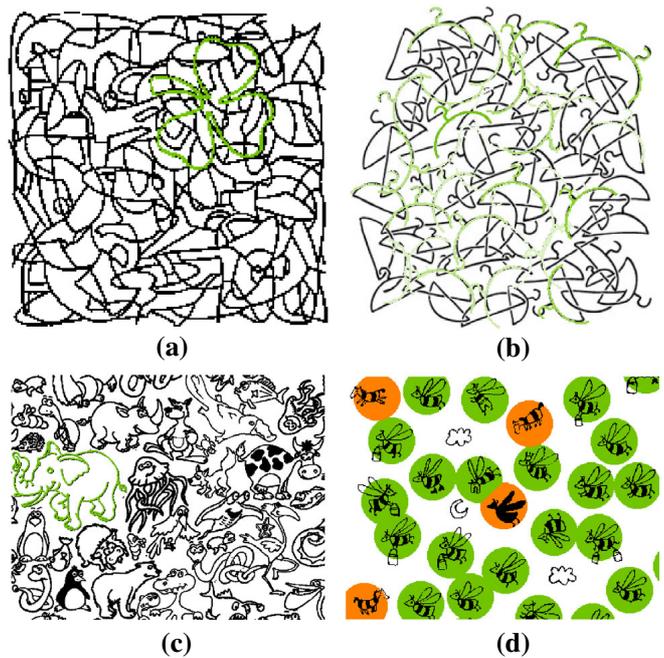
**Fig. 31** Hypotheses for pi weighted by their matching costs



**Fig. 32** A yet hidden symmetry appears when using the target figure at the *top left* in Fig. 24

Figure 30a shows the target figure, pi, detected in the line drawing Fig. 18. The algorithm determined the location where pi obtained the best matching cost. Hypotheses of 15 runs, weighted by the matching cost, are shown in Fig. 31. The hypothesis leading to the best cost is thresholded and the binary shape is depicted in blue on the illustration and shown as a closeup in Fig. 30c. Indeed, the letter 'pi' is hidden in the line drawing.

Another option is to search for a cutout of the drawing, like e.g., the segment shown top left in Fig. 24. The results of 100 independent runs searching for this segment are illustrated



**Fig. 33** Desired results. The proposed algorithm is able to localize the target figures. **a** Hidden cloverleaf. **b** Several hangers. **c** Hidden elephant. **d** Multiple bees

in Fig. 32. The collection of butterfly locations in Fig. 25 already revealed the outer circular structure of the pattern. The collection of these results leads to an emergence of a diamond scepter (as common in a mandala) together with a weak inner circular arrangement. A yet hidden symmetry appears.

## 6 Summary and Conclusion

We addressed the task of tracing out target figures in sketch-like binary teeming figure pictures. Some results of our algorithm are summarized in Fig. 33. We can search for the unique occurrence of a target figure (left column) as well as for various similar objects (right column). Particularly suited to the task, we propose a simple heuristic for generating diffuse drawings that imitate curvature coding distance images which are typically computed as solutions to elliptic PDEs. Our work extends the applications of diffusion based ideas to an interesting problem.

**Acknowledgments** We thank three anonymous reviewers for their constructive feedback. We also acknowledge the financial support from the Alexander von Humboldt Foundation and Tubitak through Grant 112E208.

## References

1. Ambrosio, L., Tortorelli, V.M.: Approximation of functional depending on jumps by elliptic functional via t-convergence. *Commun. Pure Appl. Math.* **43**(8), 999–1036 (1990)

2. Aslan, C., Tari, S.: An axis-based representation for recognition. In: IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 1339–1346 (2005)
3. Aubert, G., Aujol, J.F.: Poisson skeleton revisited: a new mathematical perspective. *J. Math. Imaging Vision (JMIV)* **48**(1), 149–159 (2014)
4. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognit.* **13**(2), 111–122 (1981)
5. Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: two new techniques for image matching. In: International Joint Conference on Artificial Intelligence (IJCAI), vol. 2, pp. 659–663. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1977)
6. Bergbauer, J., Tari, S.: Wimmelbild analysis with approximate curvature coding distance images. In: Scale Space and Variational Methods in Computer Vision (SSVM), pp. 489–500. Springer, Berlin (2013)
7. Blake, A., Zisserman, A.: Visual Reconstruction, vol. 2. MIT press, Cambridge (1987)
8. Chu, H.K., Hsu, W.H., Mitra, N.J., Cohen-Or, D., Wong, T.T., Lee, T.Y.: Camouflage images. *ACM Trans. Gr.* **29**(4), 51 (2010)
9. Fornasier, M., Toniolo, D.: Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images. *Pattern Recognit.* **38**(11), 2074–2087 (2005)
10. Gottschaldt, K.: Über den Einfluss der Erfahrung auf die Wahrnehmung von Figuren. *Psychol. Forsch.* **8**(1), 261–317 (1926)
11. Gurumoorthy, K.S., Rangarajan, A.: A Schrödinger equation for the fast computation of approximate Euclidean distance functions. In: Scale Space and Variational Methods in Computer Vision (SSVM), pp. 100–111. Springer, Berlin (2009)
12. Keles, H., Ozkar, M., Tari, S.: Weighted shapes for embedding perceived wholes. *Environ. Plan.* **39**, 360–375 (2012)
13. Ma, T., Yang, X., Latecki, L.J.: Boosting chamfer matching by learning chamfer distance normalization. In: European Conference on Computer Vision (ECCV), pp. 450–463. Springer, Berlin (2010)
14. Mainberger, M., Schmaltz, C., Berg, M., Weickert, J., Backes, M.: Diffusion-based image compression in steganography. In: International Symposium on Advances in Visual Computing (ISVC), pp. 219–228. Springer, Berlin (2012)
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
16. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer, Berlin (2003)
17. Paragios, N.: A variational approach for the segmentation of the left ventricle in cardiac image analysis. *Int. J. Comput. Vis. (IJCV)* **50**(3), 345–362 (2002)
18. Tari, S., Genctav, M.: From a modified Ambrosio-Tortorelli to a randomized part hierarchy tree. In: Scale Space and Variational Methods in Computer Vision (SSVM), pp. 267–278. Springer, Berlin (2012)
19. Tari, S., Genctav, M.: From a non-local Ambrosio-Tortorelli phase field to a randomized part hierarchy tree. *J. Math. Imaging Vis. (JMIV)* **49**(1), 69–86 (2014)
20. Tari, S., Shah, J., Pien, H.: Extraction of shape skeletons from grayscale images. *Comput. Vis. Image Underst.* **66**(2), 133–146 (1997)



Julia Diebold received her Bachelor of Science in Mathematics (2010) and her Master of Mathematics in Science and Engineering (2012) from the Technical University of Munich. She received the Achievement Award for Master Graduate 2012 of the Women for Math Science Program at the Technical University of Munich. She is supported by the German National Academic Foundation. Since November 2012, she is a Ph.D. Student in the Research Group for Computer Vision and Pattern Recognition at the Technical University of Munich, headed by Professor Daniel Cremers.



Sibel Tari is a full professor of Computer Engineering at Middle East Technical University. She received her interdisciplinary Ph.D. degree in 1997 from Northeastern University under the supervision of Jayant Shah, and her B.S. degree from the Hacettepe University Computer Science and Engineering Department in 1989. She has been with the Middle East Technical University since 1998 where she has organized and taught courses on Dynamic Systems and Feedback, Shape Analysis, and Computer Vision.



Daniel Cremers received the MS (Diplom) degree in theoretical physics in 1997 from the University of Heidelberg, Germany, and the Ph.D. degree in computer science in 2002 from the University of Mannheim, Germany. Subsequently, he spent two years as a postdoctoral researcher at the University of California Los Angeles (UCLA) and one year as a permanent researcher at Siemens Corporate Research in Princeton, New Jersey. From 2005 to 2009, he headed the Computer Vision Group at the University of Bonn, Germany. Since 2009, he has been a full professor of computer science and mathematics at the Technische Universität München, Germany. In December 2010 he was listed among "Germany's top 40 researchers below 40" (Capital).