

A robust object category detection system using deformable shapes

Robert Smith · Binh Pham

Received: 30 October 2006 / Revised: 12 July 2007 / Accepted: 29 August 2007
© Springer-Verlag 2007

Abstract An object can often be uniquely identified by its shape, which is usually fairly invariant. However, when the search is for a type of object or an object category, there can be variations in object deformation (i.e. variations in body shapes) and articulation (i.e. joint movement by limbs) that complicate their detection. We present a system that can account for this articulation variation to improve the robustness of its object detection by using deformable shapes as its main search criteria. However, existing search techniques based on deformable shapes suffer from slow search times and poor best matches when images are cluttered and the search is not initialised. To overcome these drawbacks, our object detection system uses flexible shape templates that are *augmented* by salient object features and user-defined heuristics. Our approach reduces computation time by prioritising the search around these salient features and uses the template heuristics to find truer positive matches.

Keywords Shape detection · Deformable templates · Object recognition

1 Introduction

The search for objects within images is essential for the automated understanding of the image content. Our aim is to identify the relative locations of objects in images so that a semantic interpretation of the image content can be made.

Objects can be identified by using combination of various elements—colour, texture, shape, and other distinguishing primitive features. An example is a rooster, which might be identified by using any of the following—its outline or shape (see Fig. 1), its colours, the texture of certain parts (e.g. feathers), its position relative to other objects or the ground, and other constituent features such as its eyes, beak or feet.

Some object detection systems use machine learning techniques such as support vector machines (SVMs) [12, 14], boosting [18, 16] and neural networks [6]. While fast and somewhat robust, they are difficult to initialise and train. Training can require thousands of positive examples to be gathered and depending on the feature set, can take days or even months of runtime to complete [16]. Furthermore, they cannot handle variations in rotation or object articulation. Instead we would like to use deformable shapes as our major search criteria—augmented and guided by other primitive features.

We focus on an object's shape as the main distinguishing attribute because it is generally considered the most invariant of all its features [15] (with some exceptions such as fruit, where colour is a strong invariant). However, shape can be very difficult to search for with variations in scale and rotation making the search space in an image quite large. Variations in possible object deformation and articulation then further exacerbate the problem.

Prior work in deformable shapes, namely by Felzenszwalb [4] and Chang et al. [2], have produced good results for identifying flexible shape templates in images. Their approaches differ from contour or snake models [8, 19], and approaches using “geometric hashing” [10, 11, 15], because they use shape representations that encode the internal structure of the object. The representation of the shape is formed using a triangulated polygon—which enables the template placement

R. Smith (✉) · B. Pham
Queensland University of Technology,
126 Margaret St, Brisbane, Australia
e-mail: r2.smith@qut.edu.au

B. Pham
e-mail: b.pham@qut.edu.au



Fig. 1 The silhouette of a rooster describing its shape

and deformation to be more flexibly controlled by each triangle region within the template. However, a drawback is the long search time when using complicated polygons in large search spaces. Furthermore, there are no mechanisms to guide or initialise the search of cluttered images, in which image noise can distract the template fitness functions and produce spurious results.

To overcome these problems, we incorporate additional knowledge about the object categories into the deformable templates in order to perform more efficient and robust object detection. Our system improves shape detection in two ways. First, it prioritises the search for certain shape template vertices around salient features found in the image. Second, it incorporates specific knowledge about the possible deformations of the shape templates and the relative importance of various model parts in order to further refine the results. These enhancements improve both the search times and results.

Our present application domain is traditional Vietnamese folk paintings where object shapes are the most invariant feature because the artworks are usually quite simplistic in style and colour palettes. However, in this paper we also test ordinary digital photos to show how our system performs on typical real-life images. The types of objects we currently identify are people, animals, and some more rigid objects such as musical instruments. These objects can have large variations in articulation and so require techniques that can flexibly find these variations but without being burdensome to define and manipulate.

The aim of this paper is to present our new augmented deformable shape template and detection system. Section 2 discusses the advantages of deformable models, how they are represented and how existing algorithms use them to search images. Section 3 then outlines the limitations and drawbacks of existing techniques, and Sect. 4 presents our system to overcome these problems. The experimental results and outcomes are discussed in Sect. 5 while Sect. 6 provides our conclusion and future work.

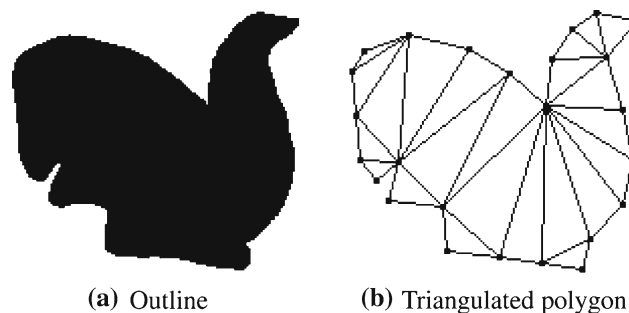


Fig. 2 The original outline and triangulated polygon template

2 Deformable shape matching

Our investigation is motivated by the deformable shape matching techniques first presented by Felzenszwalb [4] and later used by Chang [2]. The approach requires a shape template that describes the approximate outline of the object, and searches an image to find a global optimal solution for the match. Our system also uses triangulated shape templates as its main input for searching an image but is augmented by identifying salient features in the image that are associated with the object (described in Sect. 4).

2.1 Shape representation

A deformable shape model can be represented by a simple silhouette defined completely by its bounding outline (Fig. 2a). It has a simple and compact representation and is easy to use, with relatively little set up and configuration time. The silhouettes or outlines are also easy to create with any drawing software such as Gimp [7] and Dia [3].

The outline is triangulated using constrained delaunay triangulation (CDT) [1] for use as the input to the image search (Fig. 2b). The advantage in using CDT is that the object model is decomposed into natural parts, with diagonals always cleanly separating the extremities of the model (such as limbs and other protrusions). For example, Fig. 2b shows how the neck at all times throughout the triangulation is clearly delineated from the body and the head.

2.2 Shape search

In general, the shape template vertices are located on the image while optimising their placement according to a function which determines the cost of placing each triangle and edge from within the triangulated shape template onto the image.

In order to reduce the search space and time, the image is firstly overlaid with a coarse search grid. To search every pixel point of the image would usually be computationally prohibitive. Instead, we can place a 20×20 search grid over



Fig. 3 A 20×20 search grid over a picture of a boy holding a rooster

the painting of a small boy holding a rooster (Fig. 3) to effectively reduce the search space to the grid size, which in this case is now only 400 grid points. This substantially reduces the search time, although it can make the template placement less accurate since the vertices must be placed at the grid points. This is only usually a problem when the vertices of the shape template are much more finely spaced than they appear in the search grid.

The optimised placement of the template model is then calculated according to the cost function (as described in the next section). The lowest cost is returned as the best match—such as the one found in Fig. 4.

2.3 Typical cost functions

The core cost function has two parts: the *fitness cost* and the *deformation cost*. There are many different ways to define these parts—those used in the original implementation by Felzenszwalb [5] are described here so we can extend them later in Sect. 4.

The fitness cost The fitness cost is a measure of the correspondence between the mapped features and a prominent feature in the image. In the original implementation, it is the component of the image gradient that is perpendicular to the shape boundary of the polygon. This cost can be broken into the integral for each boundary edge (linking two vertices) in the polygon.

The deformation cost The deformation cost is a measure of the difference between each triangle in the original template, to the mapping in the image. For every triangle



Fig. 4 The rooster found in the image

in the shape template, we associate a cost induced by the deformation between the shape in the original template and the shape that is placed in the image. Felzenszwalb [5] uses the *log-anisotropy* cost for this calculation.

For each triangle t in the triangulation T , there is a corresponding affine map f_t to the image I . If B is the set of boundary edges of T , and the corresponding edge mapping is f_b , then the total costs for a mapping g , relative to an image, can be written as:

$$E(g, I) = C_{Def}(T) - C_{Fit}(T) \quad (1)$$

where

$$C_{Fit}(T) = \sum_{b \in B} \int \frac{\|\nabla I \circ f_b(s) \times f'_b(s)\|}{\|f'_b(s)\|} ds \quad (2)$$

and

$$C_{Def}(T) = \lambda \sum_{t \in T} def(f_t)^2 \quad (3)$$

where $def(f_t)$ is the log-anisotropy of f_t and $\|\nabla I \circ f_b(s) \times f'_b(s)\|$ is the component of the image gradient that is perpendicular to the shape boundary at $f_b(s)$.

The deformation cost is zero when f_t is a similarity transformation. It is scaled by a parameter λ so the user can increase or decrease the importance of the deformation cost. The fitness cost is divided by $\|f'_b(s)\|$ to make it scale invariant and it is *deducted* from the deformation cost to attract the template boundary to locations in the image that have high

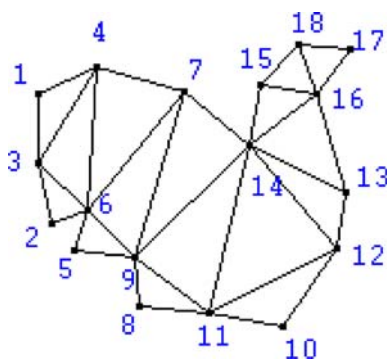


Fig. 5 A perfect elimination scheme

gradient magnitudes. The total cost function is then minimised using a dynamic programming approach described in the next subsection.

2.4 Dynamic programming

The technique used to find the lowest cost for placing the template within the image is called non-serial dynamic programming. This approach reduces the search to polynomial time from what would otherwise be exponential time using a naive, brute force approach. In order to apply this optimisation technique, we require a chain structure or more specifically, a decomposable graph that does not have large cliques. Such is the case with the triangulated polygon of the input template (Fig. 2b)—it is a dual graph with several such chain structures called *perfect elimination schemes*. These schemes provide an elimination order of the vertices from the polygon such that when eliminating the i th vertex, it can be found in only one triangle of the current triangulated polygon. An example of such a scheme for the triangulated bird outline is in Fig. 5. If we remove the vertices from the polygon in the labeled order, each vertex when it is removed, is only ever in one triangle. Thus, after each removal, the remaining part is still a triangulation from within the original polygon.

The algorithm for non-serial dynamic programming sequentially eliminates the vertices of the triangulated polygon and stores the cost of placing that vertex as a function of the other two vertices in the triangle. The process has been developed by Felzenswalb [5], where a complete description of the original algorithm is available.

3 Problems with previous approaches

The algorithm in the previous section suffers from two main problems in practice—the slow search time due to the computational complexity (even using dynamic programming); and poor results in cluttered images due to fitness costs that

are too broadly defined and inflexible to find the right features that belong to the search template.

3.1 Slow image search

The algorithm described in Sect. 2.4 runs in $O(nm^3)$ time, where n is the number of vertices in the polygon and m is the number of possible locations for each vertex. As the number of possible grid points increases, the time taken to run the algorithm can become prohibitively slow.

Various heuristics can be applied to reduce the search space and time. One example is to only search near the “ideal” third vertex of any triangle once two have been chosen [5], reducing the order to $O(nm^2)$. Even using this improvement, using a P4 computer (3 GHz, 1 GB RAM) to search a 50×50 grid (2,500 grid points) for a shape represented by a polygon with 25 vertices, can still take the search algorithm up to 3 min to complete. This time frame restricts the search technique to applications that are completely off-line. Furthermore, since the search is also dependent on the tuning of various parameters and thresholds, the lengthy search time can make the initial configuration very frustrating.

A second heuristic limits the invariance to scale by only placing triangles that are 80–120% of the size of those in the original template [2]. Together these heuristics can reduce the complexity to $O(nm)$, but also reduce the scope of potential matches and is not appropriate for all domains.

3.2 Poor “best” matches

The best match (i.e. with the lowest cost) in cluttered images is often wrong due to the fitness function having difficulty in identifying the relevant salient features—potentially attracting the template to spurious features. In Fig. 6c, the deformable lute template (Fig. 6b), is more attracted to the larger gradient differences found in the body and hair regions of the image than around the lute itself. The fitness cost function is unable to identify the appropriate salient features because it applies the same function to all vertices without any importance weighting. The single balancing parameter λ , that weights the importance between shape and fitness, does not provide sufficient flexibility for guiding the placement of the vertices.

Another reason for some poor results, is that there is no mechanism to identify important features in the images for a particular template—all parts of the templates must be assessed with the *same* fitness cost function. This means that templates cannot make use of salient object features, such as the texture of a surface or a simple geometric shape always present in the template (e.g. the wheels on a car). These features could be associated with certain parts of the template.

These sometimes poor results together with the slow search times can limit the application of this technique in

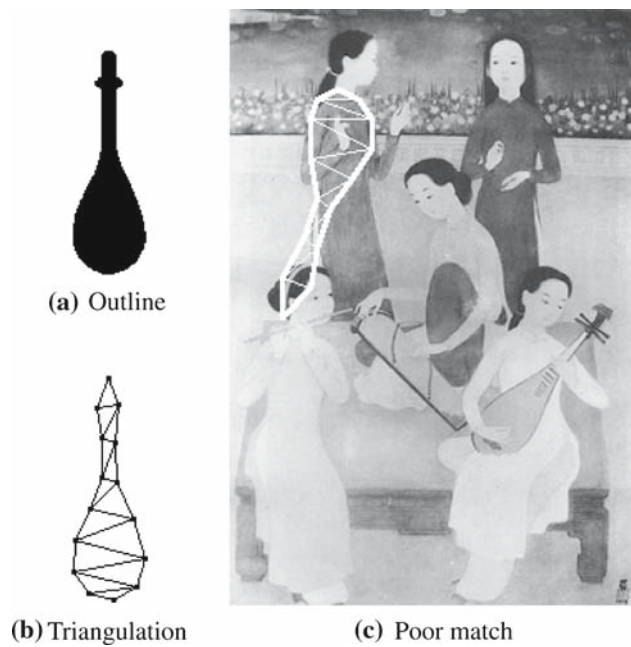


Fig. 6 Lute not found in image

images with cluttered scenes or poorly contrasted objects. We present techniques to overcome these limitations in the next section.

4 New shape template and search system

The main idea behind our approach is to incorporate additional information about the object being sought into the deformable shape template, which we then call the *augmented shape template*. This extra information is used to firstly find *salient features* within the image, which are then used to apply additional costs to the position of certain *priority vertices* relative to these predetermined salient features. Salient features can be colours, geometric shapes, or textures—practically anything that the user might like to define to augment the shape template.

Our system introduces a new term to the cost function called the *deviation cost*. This is a cost to place the priority vertices away from the position of the salient features in the image. The cost to place a priority vertex precisely at its associated salient feature is zero. Input images are preprocessed to identify these salient features, which are then used as additional inputs cost function of the search algorithm. This cost is described in Sect. 4.2.

Another improvement is in making the fitness costs configurable when placing each vertex of the template. We have found that by allowing the fitness costs to be flexibly weighted, we can find matches in difficult images that would otherwise be unachievable. This enhancement is discussed in Sect. 4.3.

Similar to the fitness costs, the deformation costs have been made flexible for every triangle comprising the template. This has already been used by Felzenszwalb to compute the deformable cost weightings for template triangles from aggregating various input images [5]. It is important to incorporate this enhancement into our model as well because it allows much more flexibility when matching objects that have areas of variable flexibility (e.g. an arm or leg). This improvement is discussed in Sect. 4.4.

4.1 Salient feature extraction

Various feature detectors are activated by the search system according to the salient features are required by the augmented shape template. Software components that calculate these features can be switched on as necessary by the system by inspecting the shape model. These detectors usually have a very small processing cost associated with them—colour and edge detection for example. The new search can only apply priority vertices to the image if the appropriate salient features were found. The positions of these features in the image are then used as an input to our new deviation costs, outlined in the next subsection.

Some examples of salient features that can guide the image search are as follows:

- *Edges*: image edges can be used to constrain the extent of the search for each vertex of the template (images can then be searched much more quickly as shown in the example below);
- *Same edges*: some of the template vertices might be restricted to lying on edge lines that are linked;
- *Simple geometrical features*: circles, rectangles, corners and lines (intersecting or parallel) can all help localise the shape template;
- *Colours and texture*: a distinctive colour or texture that normally distinguishes the object can easily be a salient feature;
- *Keypoints*: distinctive keypoints that may be detected using specialised techniques such as PCA-SIFT [9];
- *Other flexible templates*: searching for other templates that are also deformable but are only a small part of the larger shape being sought can be done quickly and used to help localise certain shape vertices;
- *Other classifiers*: if already available, statistical classifiers (e.g. face or object detectors) can be used to quickly identify some salient features.

In fact, image edges (the first salient feature listed above) were used by Chang [2] to reduce their image search space. However, this is only one special case out of many possibilities such as already listed. We have abstracted this approach

to formalise a methodology that can incorporate *any* salient feature that may be relevant to the shape template.

4.2 New deviation costs

The augmented shape template is annotated with *priority vertices* that specify the points in the shape that should be associated with a salient feature in the image. Every vertex in the shape template could be associated with a salient feature to restrict the search space, thereby reducing the search time *and* improving the results by eliminating many potential bad matches.

The *deviation cost* captures the cost of placing a priority vertex some distance from its associated salient feature. We add a third term to the cost function back in Eq. 1 as follows:

$$E(g, I) = C_{\text{Dev}}(T) + C_{\text{Def}}(T) - C_{\text{Fit}}(T), \quad (4)$$

where

$$C_{\text{Dev}}(T) = \begin{cases} \sum_{p \in P} \theta_p \text{dist}(p, f) & \text{for } p \in \text{neig}(f) \\ \infty & \text{for } p \notin \text{neig}(f), \end{cases} \quad (5)$$

where θ_p is a deviation cost weighting parameter for each priority vertex, $\text{dist}(p, f)$ is the distance cost from a priority vertex position p to the associated salient feature position f (e.g. a function of Euclidian distance), and $\text{neig}(f)$ is the function determining the neighbourhood of f .

4.2.1 An example

A simple example can be made using a Canny edge detection function to identify edges to be used as salient features. All vertices can be associated with this salient feature so that the deviation cost is zero for vertices placed at or near edge points, and infinite when placed more than one grid point away. The search is of course discontinued once the cost becomes too high. Figure 7 shows an image with grid points removed that cause a high placement costs. The resulting search grid only has 212 points (compared with 400 in Fig. 3), reducing the search time to approximately 27% of the original time—with a similar best match.

The resultant shape outline can vary slightly but from the one found using the entire grid, but since it is only an approximation based on the original silhouette, this is usually not a problem.

4.3 Configurable fitness costs

For every vertex placement in the image, we now also allow an adjustable weighting to be used, which determines how important the fitness cost for this particular vertex placement



Fig. 7 Grid points resulting in a high deviation cost have been omitted

is. They may all be equally important, as with the original algorithm, or they may vary according to the object's requirements. Certain areas or edges of the shape template can now be given a higher or lower weighting when matching. This makes it possible to include heuristics to make some parts of the model more or less important than other parts—for example the beak of a bird may be far more important match using the fitness function than perhaps the feathers or the feet.

We redefine the fitness cost function in Eq. 2 as follows:

$$C_{\text{Fit}}(T) = \sum_{b \in B} \kappa_b \int \frac{\|\nabla I \circ f_b(s) \times f'_b(s)\|}{\|f'_b(s)\|} ds \quad (6)$$

where κ_b is the fitness weighting parameter for each boundary edge b .

4.3.1 An example

This improvement can be very useful when some parts of the model are more important to match and need to be weighted accordingly, or conversely, when a part of the model has a boundary edge that should have no significant effect on the fitness cost. A clear example would be an silhouette that is a part of the greater object—such as a head and shoulder model which is only part of the person (Fig. 8a). The result when searching for this template with no fitness function adjustments is in Fig. 8c. However, the bottom edge of the model should not contribute at all to the fitness cost because it does not represent a true termination of the object boundary (i.e.

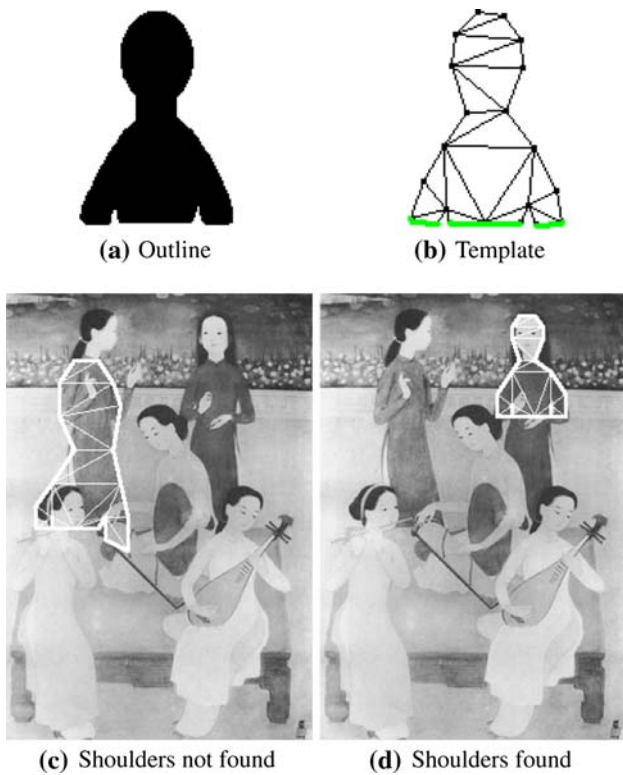


Fig. 8 Configurable fitness costs enable partial objects to be found

the person does not end there). When we set $\kappa = 0$ for the bottom boundary edges of the triangulated model (Fig. 8b) a true positive match is found (Fig. 8d).

4.4 Configurable deformation costs

The deformation cost in the original cost function is balanced by a parameter λ . This parameter is applied to the entire deformation cost term, i.e. it applies to all triangles equally. However, it is better to allow the deformation weighting to be individually specified for every triangle. Felzenszwalb [5] uses this approach to learn deformation parameters from training data and suggests the deformation costs for individual triangles could be tuned. We have also explicitly incorporated this parameter into our model specification and redefine the deformation cost function in Eq. 3 as follows:

$$C_{\text{Def}}(T) = \sum_{t \in T} \lambda_t \text{def}(f_t)^2, \tag{7}$$

where λ_t is the deformation weighting parameter for each triangle t .

4.4.1 An example

The effect of this flexible configuration can be easily demonstrated by referring to the template model in Fig. 9a with the

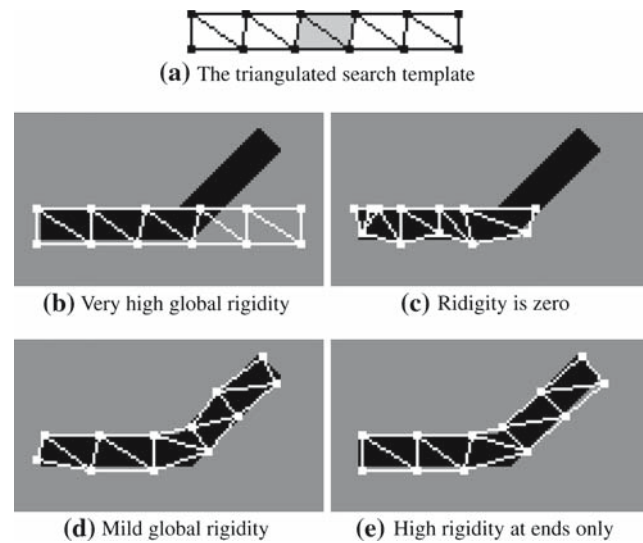


Fig. 9 Variable deformation costs improve the accuracy of the match

triangulation shown. All triangles have a uniformly applied deformation weighting at first. When searching using a highly rigid global weighting (i.e. $\lambda = 100$), the template cannot deform at all to match the bent bar in the image (Fig. 9b). If there is no deformation cost at all ($\lambda = 0$), then salient features dominate the search and the result is as seen in Fig. 9c. When the template rigidity is mild to moderate ($\lambda = 40$), the best match is shown in Fig. 9d.

However, a better approach is to define the local areas of high flexibility in the triangulated model. In Fig. 9e, the template has been configured to allow high flexibility in the shaded triangles (in Fig. 9a) and to be quite rigid otherwise. Of note is that the total cost to match Fig. 9d and e is exactly the same, but only the middle, shaded triangles have been deformed in the latter. Shape models that include adjustable deformation weightings are better able to represent the true nature of the object (such as a hinge, flexible joint or stretchable area).

4.5 New search system

An overview of the system can be seen in Fig. 10. The new augmented shape templates are composed of four parts: the triangulated shape outline (black); its priority vertices and their associated salient features (red); the fitness cost weightings (green); and the deformation cost weightings (grey). These form the inputs to the three parts of the cost function—the deviation, fitness and deformation costs.

The priority vertices of the shape template specify which feature detectors are activated and used by the system. These determine the salient feature locations in the image and are used to determine the deviation costs of the priority vertices. The deformation and fitness weightings are applied to the

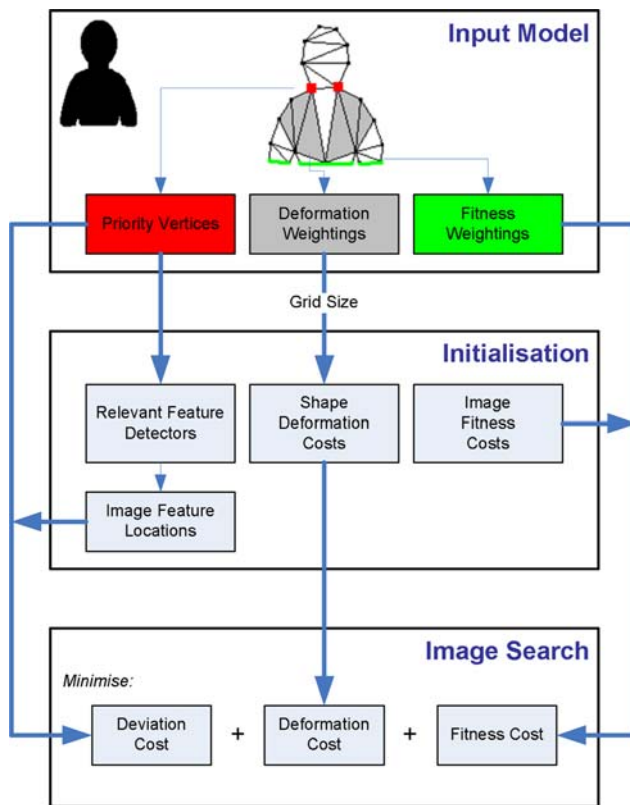


Fig. 10 The system overview

costs for each of the triangles and boundary edges respectively.

In effect, each placement of a priority vertex in the shape template is on a grid with a variable number of grid points that depend on the salient features of the image. If there are few salient features for that vertex, then the number of grid points to search will also be few. Therefore, while the total cost function is still of complexity $O(nm^2)$, the priority vertices have associated grid sizes (m) that are often drastically reduced and this is how good reductions in run-time are achieved.

5 Experimental results

We present two examples to demonstrate the system using the improved cost function. By using a combination of priority vertices with salient features, configurable fitness costs and configurable deformation costs, we have been able to improve the match results *and* the running time of the search using this system. The first example is in the domain of traditional Vietnamese folk paintings (improving the result seen back in Fig. 6) and the second is using a typical camera photo image.

In the first example, we took the original lute template from Fig. 6a and used two salient features to create priority

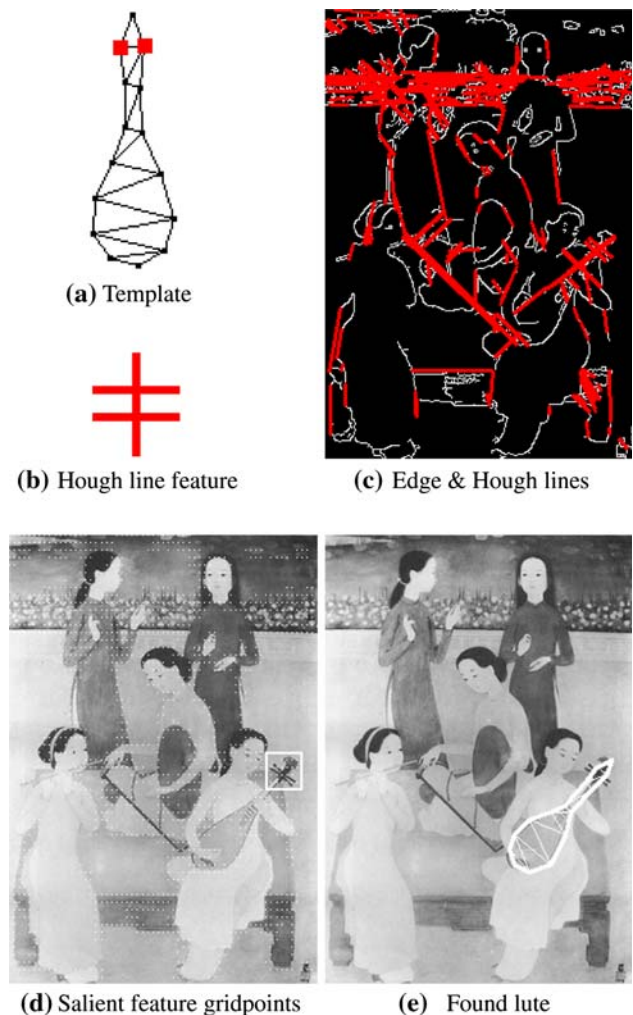


Fig. 11 Salient features used to refine the search for the lute

vertices—edges and Hough lines. All vertices are associated with the edge feature and two special priority vertices (marked in red in (Fig. 11a) are associated with a Hough feature—specified as two nearby parallel lines intersected perpendicularly by a third (Fig. 11b). This is a feature always found at the tuning area of the lute and so is used to augment the shape template.

In the initialisation stage, we identify the edges and probabilistic Hough lines of the image (Fig. 11c). This will effectively reduce the search space for the priority vertices to the salient feature grid points (Fig. 11d) because the deviation cost is ∞ for vertices which are more than one grid point away from edges or more than a 3×3 square grid away from the hough line features. The resultant search outcome is in Fig. 11e. It a successful match and the time taken to search this image has been reduced from 81 to 51 s (including the fixed time costs of the initialisation stage).

The second example of the improved search system using a photo image is shown in Fig. 12. In this image, two

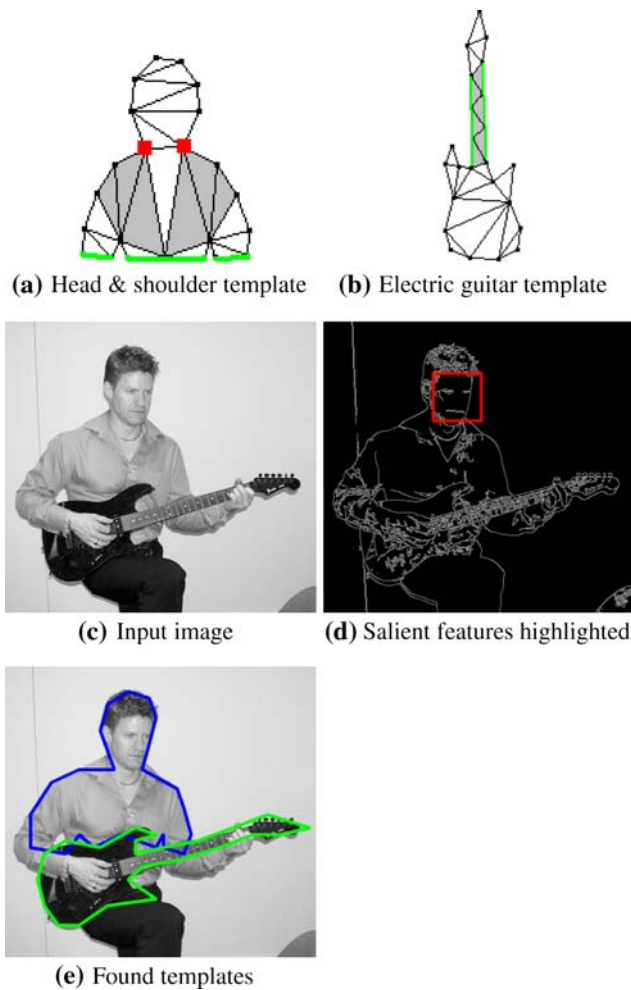


Fig. 12 Two templates found using salient features

templates were searched for—a head and shoulders template (Figure 12a) and an electric guitar template (Fig. 12b). The head & shoulders template is for a man so the shoulders are initialised to be broader than in Fig. 8b. The cost of placing priority vertices of the neck (in red) are weighted (using θ_p) by the distance to the salient features found by a face detector (as in OpenCV [13] which implements a version of the statistical facial classifier designed by Viola and Jones [17]). The lower left and lower right corners of the face bounding box are associated with the left and right priority vertices, respectively (Fig. 12d). Furthermore, *all* vertices are associated with the image edges (also shown in Fig. 12d). Next, the fitness cost weighting (κ_b) for the edges at the bottom of the template (in blue) are made 0—because this is not a true termination of the partial object. Lastly, while the entire template is set to be somewhat flexible ($\lambda = 30$), the shoulder areas (the shaded triangles) are set to be very flexible ($\lambda_t = 1$).

The guitar, being a relatively rigid object, is given a higher global rigidity setting ($\lambda = 60$) applying to every triangle. The

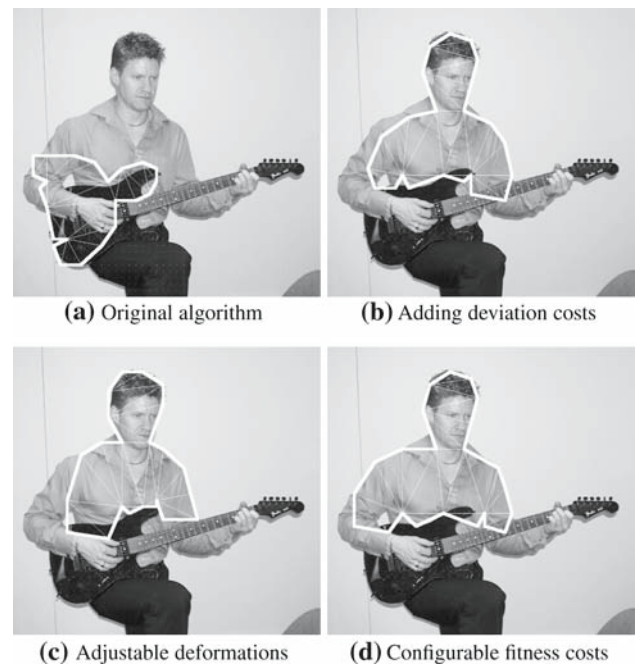


Fig. 13 Search results after each of our algorithm improvements (shown cumulatively)

boundary edges along the neck (shown in blue) are weighted with a low κ_b so that there is no tendency to deviate when it is occluded by the hand that is playing chords. Furthermore, the triangles comprising most of the neck are made even more rigid ($\lambda_t = 100$) since this area should be inflexible apart from some variation in length (allowed by the top three triangles of the neck). Lastly, all vertices of the the guitar are again associated with image edges. The results are shown in Fig. 12e.

Both matches are good results considering the coarseness of the shape polygon used in these templates. A finer template (i.e. with more triangles) could be used to improve the outlines, but with more computational expense.

5.1 Improved search results

The improved matching results with the new algorithm are significant. Not only does it take less time to search (see the next subsection), but the results are much better. The improvements can be seen in Fig. 13 as we introduce each of our cost function enhancements one at a time.

Figure 13a shows the best match result from the original algorithm. This match took 27 s to complete and is obviously not a match. Figure 13b shows our result after the inclusion of the priority vertices—the result is already much better and the time taken is 8 s. Figure 13c and d show the inclusions of the variable deformation costs and adjustable fitness costs, respectively—with the search time still taking 8 s. The result improves with each enhancement to the cost function.

Table 1 Timing results for the head and shoulders template search in Fig. 13a compared with Fig. 13d

Process	Orig. (s)	New (s)	Reduction (%)
A. Deformation costs (fixed)	23	23	0
B. Image fitness costs	4	5	-25
C. Image grid search	23	3	87
Image search (B + C)	27	8	65

With the new cost function we can far more flexibly define the input shape template and associate heuristics to form a better input model. The new deviation costs effectively apply prohibitive costs to improbable matches thereby excluding them from the search space.

5.2 Shortened search times

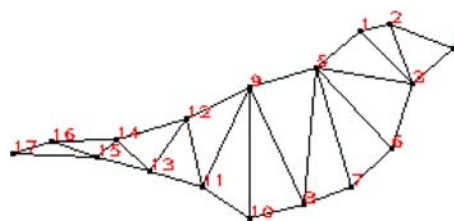
In Table 1, we compare some typical timings of the original algorithm against our new one incorporating deviation costs, and adjustable fitness and deformation costs. These results are for matching of the head & shoulders template to the man playing the guitar.

The deformation cost calculation time remains unchanged because these costs are specific to the shape template and grid size. They are not dependant on the image. With more memory, these costs could be stored between searches and not incurred after the first search with a template. The dramatic reduction in iteration search time (from 23 to 3s) is due to the reduction in searched grid points—from 2,025 to 618. This will not always be the case and will depend on the salient features. For example, when using edges as a salient feature, cluttered images will require more grid points to be searched—but time improvements should still be noticeable.

6 Quantitative performance evaluation

In this section we evaluate the performance improvement of the new algorithm and framework over the original algorithm, using a batch of bird images with varying clutter. Firstly, we use the original algorithm and a simple shape template (Fig. 14) to search for a bird in the series of bird images (Fig. 15). The only enhancement is that we use image edges as a salient feature to greatly improve the search time. It can be seen that the original algorithm fails to detect the bird as the images become more cluttered and complicated.

The shape template is then augmented to be more flexible through the “body” area. This allows the template to accommodate a greater variety of bird body types. The template also requires points 1 and 3 to be near salient features—in this case Hough circle centres found using OpenCV. This is

**Fig. 14** The simple shape template for a bird

an attempt to detect the eye and head of the bird and guide the “head” to be near it. This augmented template can be seen in Fig. 16 which we use to improve the search results (shown in Fig. 17).

Because we are detecting the object outlines, there is no binary outcome in terms of a success or failure. Most often there is in fact a partial match. So in order to evaluate the performance improvement we calculate the average distance from the detected polygon shape to the ground truth boundaries for each image using the search grid as a reference (grid points are equally spaced in all images). In this way we can capture the result of partial matches (whereas receiver operating characteristics would not). The average misalignment error and search times are shown in Table 2.

To summarise, the new framework is using the salient features (edges and hough circles) to improve the detection quality and reduce search times. In fact, if edges had not been used in the first instance, then the average search time would have been even higher at 54 s with no better detection quality.

7 Conclusions and future work

Deformable models are a powerful and elegant way to search for entire categories of objects within images. To address the drawbacks of deformable shape searching, we have developed a system for object category detection that is more robust than simpler deformable model search techniques.

Significant improvements in speed and match results were achieved by incorporating object specific heuristics into the deformable templates. These new augmented shape templates contain associations with salient image features to guide the search, and weighting information for variable deformation and fitness costs. The system produces better object matches and search times are often drastically reduced. Parameter tuning requires some effort for a template in a new search domain—but once the template has been successfully configured, the system performs quite robustly in that domain. The salient feature detectors are still under development and improvements will be sought using Hough lines and circles.



Fig. 15 Search results for the original algorithm

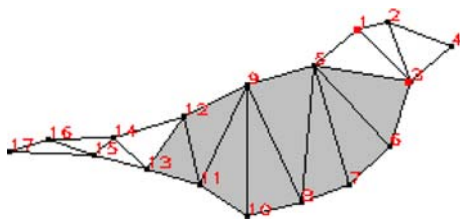


Fig. 16 The augmented shape template for a bird with more flexible areas shown in grey and priority vertices at points 1 and 3

Our eventual aim is to enable the semantic interpretation of image content. To this end, Fig. 12e is a perfect example of how the location of two objects in an image can be used to infer the semantics of the image—which in

Table 2 Average misalignment error (measured by gridpoints) and search time comparison between for the old (Fig. 15) and new (Fig. 17) approaches

	Orig.	New	Reduction (%)
Average error (grid distance)	8.51	3.37	60.4
Average time taken (s)	22.9s	19.1s	16.6

this case might be “man playing electric guitar”. The relative articulation and location of each template found can give important indicators of their meaning within the image.



Fig. 17 Search results for the new framework

Acknowledgments The authors wish to thank Dr. Felzenszwalb for his code on the original deformable template search. This work was made possible by the Australian Research Council Centre of Excellence for Creative Industries and Innovation. We also thank BirdWay (www.birdway.com.au) for the non-commercial use of its images.

References

- Bern, M.W., Eppstein, D.: Mesh generation and optimal triangulation. In: Du, D.Z., Hwang, F.K.M. (eds.) *Computing in Euclidean Geometry*, 2nd edn. Lecture Notes Series on Computing, vol. 4, pp. 47–123. World Scientific, Singapore (1995) <http://www.ics.uci.edu/~eppstein/pubs/BerEpp-CEG-95.pdf>
- Chang, T.L., Liu, T.L.: Detecting deformable objects with flexible shape priors. In: *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, vol. 4, pp. 155–158. IEEE Computer Society, Washington, DC (2004) Doi:10.1109/ICPR.2004.291
- Dia: (2006) www.gnome.org/projects/dia
- Felzenszwalb, P.F.: Representation and detection of deformable shapes. *CVPR* **1**, 102–108 (2003) <http://citeseer.ist.psu.edu/felzenszwalb03representation.html>
- Felzenszwalb, P.F.: Representation and detection of deformable shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(2), 208–220 (2005) Doi:10.1109/TPAMI.2005.35
- Feraud, R., Bernier, O.J., Viallet, J.E., Collobert, M.: A fast and accurate face detector based on neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(1), 42–53 (2001) Doi:10.1109/34.899945
- GIMP: (2006) www.gimp.org
- Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Comput. Vis.* **1**(4), 321–331 (1988) Doi:10.1007/BF00133570 . <http://dx.doi.org/10.1007/BF00133570>
- Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–506–II–513, vol.2 (2004). Doi:10.1109/CVPR.2004.1315206
- Lamdan, Y., Schwartz, J., Wolfson, H.: Object recognition by affine invariant matching. In: *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR '88., Computer Society Conference on*, pp. 335–344 (1988) Doi:10.1109/CVPR.1988.196257
- Li, W.J., Lee, T.: Image registration and object recognition by affine invariant matching. In: *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pp. 56–59 (2001) Doi:10.1109/ISIMP.2001.925330
- Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(4), 349–361 (2001) Doi:10.1109/34.917571
- OpenCV: (2006) <http://sourceforge.net/projects/opencv>
- Osuna, E., Freund, R., Girosit, F.: Training support vector machines: an application to face detection. In: *Computer Vision and Pattern Recognition, 1997. Proceedings, 1997 IEEE Computer Society Conference on*, pp. 130–136 (1997) Doi:10.1109/CVPR.1997.609310
- Thuresson, J., Carlsson, S.: Finding object categories in cluttered images using minimal shape prototypes. In: *SCIA*, pp. 1122–1129 (2003)
- Verschae, R., del Solar, J.R.: *A Hybrid Face Detector Based on an Asymmetrical Adaboost Cascade Detector and a Wavelet-Bayesian-Detector*. Springer, Berlin (2003)
- Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vision* **57**(2), 137–154 (2004) Doi:10.1023/B:VISI.0000013087.49260.fb
- Xiao, R., Zhu, L., Zhang, H.J.: Boosting chain learning for object detection. In: *Computer Vision, 2003. Proceedings. 9th IEEE International Conference on*, vol. 1, pp. 709–715 (2003). Doi:10.1109/ICCV.2003.1238417
- Yezzi A., J., Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A.: A geometric snake model for segmentation of medical imagery. *Med. Imaging, IEEE Trans.* **16**(2), 199–209 (1997). Doi:10.1109/42.563665

Author biographies



Robert Smith received his PhD degree in computer science from the Queensland University of Technology where he currently is a Research Fellow for the Visual Media Computing group. His research interests are in computer vision, autonomous control and machine learning.



Binh Pham is currently a Professor in the Faculty of Information Technology at the Queensland University of Technology, Brisbane, Australia, after holding the position of Director of Research for 7 years from 2000. Prior to this, she held the IBM Foundation Chair in Information Technology at the University of Ballarat from 1995–1999, and was an Associate Professor in the School of Computing and Information Technology at Griffith University from 1993–1995. She was the Founding Director of the Victorian Centre for Image Processing and Analysis (CIPAG) at Monash University from 1991–1993. Her research interests include computer graphics, multimedia, image analysis, intelligent systems, spatial and temporal data mining, and their application in diverse domains.