Scale Saliency: Applications in Visual Matching, Tracking and View-Based Object Recognition

Jonathon S. Hare and Paul H. Lewis

Intelligence, Agents, Multimedia Group
University of Southampton, Southampton, SO17 1BJ, UK
{jsh02r, phl}@ecs.soton.ac.uk

Abstract

In this paper, we introduce a novel technique for image matching and feature-based tracking. The technique is based on the idea of using the Scale-Saliency algorithm to pick a sparse number of 'interesting' or 'salient' features. Feature vectors for each of the salient regions are generated and used in the matching process. Due to the nature of the sparse representation of feature vectors generated by the technique, sub-image matching is also accomplished. We demonstrate the techniques robustness to geometric transformations in the query image and suggest that the technique would be suitable for view-based object recognition. We also apply the matching technique to the problem of feature tracking across multiple video frames by matching salient regions across frame pairs. We show that our tracking algorithm is able to explicitly extract the 3D motion vector of each salient region during the tracking process, using a single uncalibrated camera. We illustrate the functionality of our tracking algorithm by showing results from tracking a single salient region in near real-time with a live camera input.

1. Introduction

The use of saliency in computer vision has become quite widespread in recent years. Saliency is often used to provide the basis for a visual attention mechanism that reduces the need for computational resources [8, 5, 3, 4]. Historically, saliency was described by the term 'interest point detectors', but use of the term 'saliency' has come about from the large amount of psychology-based work on selective visual attention. The Scale-Saliency algorithm of Kadir and Brady [7, 6] defines salient regions within images as a function of local complexity weighted by a measure of self-similarity across scale space. The algorithm is able to locate circular patches within an image that incorporate 'interesting' features.

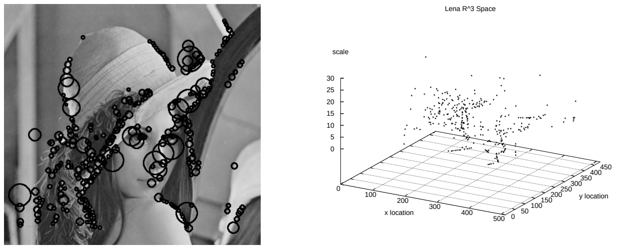
We begin with the Scale-Saliency algorithm and develop it into an image matching and object tracking facility. Historically, matching and tracking have been seen as quite different problems. However, our technique accomplishes tracking by posing the problem in terms of sub-image matching across video frames. The technique also allows a compact image signature to be created. Because the technique works on the basis of matching sub-images, it can also be applied to view-based object recognition, even when the query object is partially occluded.

2. Scale-Saliency

The Scale-Saliency algorithm developed by Kadir and Brady [7, 6] was based on earlier work by Gilles [2]. Gilles investigated salient local image patches or 'icons' to match and register two images (specifically aerial reconnaissance images). Gilles suggested that by extracting locally salient features from the pair of images and matching these, it would be possible to estimate the global transform between the two images. Gilles defined saliency in terms of local signal complexity or unpredictability. More specifically, he suggested the use of Shannon Entropy of local attributes to estimate the saliency. Basically, image segments with flatter intensity histogram distributions[1] tend to have higher signal complexity and thus higher entropy. Gilles' method only worked at a single scale, and picked salient points, rather than salient regions.

Kadir and Brady modified Gilles' original algorithm to make it perform well on images other than those from aerial reconnaissance imagery. Essentially they changed the algorithm so that it detected salient regions at multiple scales by

[1] Kadir and Brady [7] note that the method is not limited to the intensity histogram and that it is equally possible to use a histogram from a different descriptor, such as colour or edge strength



(a) (b)

Figure 1. (a)The output of Scale-Saliency algorithm applied to the Lena picture, black circles indicate salient patches with scale equal to the circle radius; (b)The corresponding $\Re^3$ Scale-Space

looking for self-similarity across scales. The modified algorithm located circular patches of the original image that were considered salient. The size of the patch was determined automatically by the multi-scale additions to Gilles' algorithm. In addition Kadir and Brady developed a simple clustering algorithm to group together features within the $\Re^3$ space that have similar x and y location, and scale. Figure 1(a) illustrates the results of applying the algorithm to an image and Figure 1(b) illustrates the corresponding $\Re^3$ space.

3. Image Matching and View-Based Recognition

We propose a technique in which image matching is performed by matching salient regions between the images. Using the Scale Saliency algorithm we generate a compact image signature based on the salient patches. This is in contrast to other recent techniques which use salient points [9].

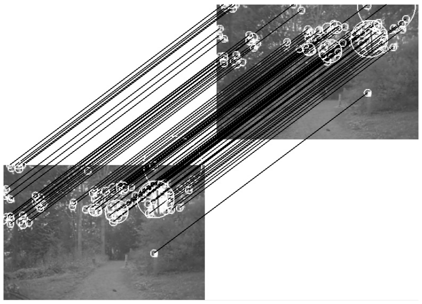


Figure 2. Matched salient regions between two images

In our current algorithm, the signature consists of a list of feature vectors, one for each of the salient regions. The feature vectors contain the spatial location (x,y), the scale, the saliency and the normalised intensity histogram of the corresponding salient patch. We decided to use the normalised intensity histogram in the matching process because of its invariance to rotation, scaling, and other geometric transformations. However, it should be noted that any region descriptor could be used for the matching. For example, a colour histogram would probably improve the matching performance of the algorithm, and could be made invariant to illumination changes, as well as geometric transformations. Our current implementation does not perform any kind of geometric consistency checking between the locations of the salient regions. As such, a matched image could have a completely different arrangement of salient features (in the $\Re^3$ space) to the query image. We hope to address this problem in the future.

To perform the actual matching between two images, A and B, the intensity histograms for each salient feature in image A are compared to each of the intensity histograms for the salient regions in image B. Figure 2 illustrates the some matched salient regions between two images. The algorithm in pseudo-code is as follows:

```
Let LA[1..N] = the list of the N normalised
histograms from image A;
Let LB[1..M] = the list of the M normalised
histograms from image B;
for i=1..N
{
Dist = min(sqrt(sqr(LA[i] - LB[1..M])))
idx = index of LB[] that gave the minimum distance
remove LB[idx]
CumDist+=dist
}
Decide on the probability of a match based on
CumDist.  A CumDist of 0 is the maximal probability
of a match.
```

The algorithm also has a potential use in content-based image retrieval based on the correspondence of features within salient regions between the query and database images. The cumulative distance measure in the algorithm is really just a measure of how different the salient regions are between the images. Thus by selecting a number of images with relatively low distances, rather than just the one with the lowest distance, a form of content based image similarity retrieval is accomplished. For example the query could be “find me 5 images with similar salient features to this [the query] image”.

3.1. View-Based Recognition

The image-matching system described above is also suited to view-based object recognition. As matching is only performed using the salient regions, it does not matter

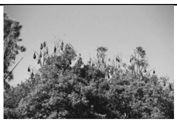
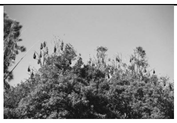
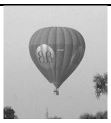
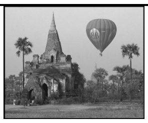
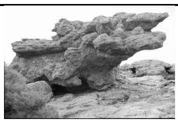
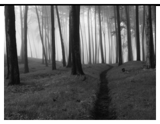
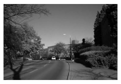
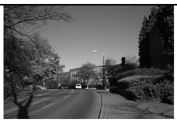
	Query	Closest Match	Match 2	Match 3	Match 4	Match 5
(a)		 Dist=0.00	 Dist=0.25	 Dist=0.27	 Dist=0.27	 Dist=0.27
(b)		 Dist=0.00	 Dist=0.34	 Dist=0.34	 Dist=0.36	 Dist=0.36
(c)		 Dist=0.40	 Dist=0.40	 Dist=0.41	 Dist=0.42	 Dist=0.42
(d)		 Dist=0.10	 Dist=0.31	 Dist=0.32	 Dist=0.32	 Dist=0.33

Table 1. Results from matching with a number of test images. Image (a) shows the matching of an image taken straight from the image database, and hence the distance is zero (exact match); Image (b) illustrates sub-image matching; Image (c) shows an attempt at matching an image that was not in the database; Image (d) shows the matching of an image taken from the database and scaled to 70% of its original size.

if the object being matched is partially occluded, as only the visible regions of the object need be matched to the database. The matching algorithm is also robust to geometric transformations of the object in the query image. Kadir [6] showed that matching using the raw pixel data from within the salient regions worked well with rotations of up to 15°. Using our matching technique improves this to in excess of 40°, at which point the salient regions tend to become unstable. Obviously, increasing the robustness of the matching algorithm to geometric transformations reduces the number of views that need to be stored in the database.

3.2. Matching Results

In order to demonstrate our system, we assembled a series of 308 images from the University of Washington Ground-Truth image database [13]. The images were then preprocessed using the Scale-Saliency algorithm to generate a feature vector database on which to test the matching algorithm. The query image then has the Scale-Saliency algorithm applied to it to generate it own list of feature vectors. The feature vectors of the query image are compared to each of the lists of feature vectors of the images in the database and the cumulative distance (`CumDist`) is recorded. The image with the smallest cumulative distance is considered the closest match.

Table 1 illustrates some results from different queries using our technique. To illustrate the algorithm's potential for content-based image retrieval, we show the 5 images with the least distance together with the query image. The results are quite interesting as they show some problems with the current algorithm, but also some potential. Query image (a) was taken straight from the database, with no alterations made to it, hence it is not surprising that it matched an image from the database exactly. Query image (b) is a cropped version of an image from the database (a sub-image). Again the match is correct. Query image (c) was not in the database and was done as an experiment to see what would happen in this case. It is interesting that all of the results for (c) are also of trees, but the distances are all quite high, telling us that the matching was relatively poor. Query image (d) is an image from the database scaled to 70% of its original size.

4. Tracking

The tracking problem can be posed as a sub-image matching problem. In our case, we want to track single salient regions across multiple frames. We use our matching technique to find matching regions between consecutive video frames. The advantage of our algorithm over some other tracking algorithms is that due to the nature of the way the Scale-Space algorithm locates salient regions in scale-space, it is possible to estimate 3D motion trajectories directly from the tracking process with a single uncalibrated camera. For example, consider a salient region corresponding to an object within an image sequence. If the object moves closer to the camera (i.e. gets bigger), then the salient region corresponding to the object will also increase in size (i.e. it's scale will increase). If the object moves further away, its scale will decrease. Thus, by tracking the salient features in the scale direction as well as in the x and y directions, 3D motion can be recovered. In addition, the algorithm does not require small inter-frame times. Our current algorithm is as follows:

```
Let LA[1..N][x,y,scale,sal,hist] = the list of the
N salient features/regions in frame 1
Let LB[1..M][x,y,scale,sal,hist] = the list of the
M salient features/regions in frame 2
for i=1..N
{
Dist = min(sqrt(sqr(LA[i] - LB[1..M])))
idx = index of LB[] that gave the minimum distance
If Dist < Thresh, a predetermined threshold
Then the motion vector in 3D is thus
LA[i][x,y,scale] -> LB[idx][x,y,scale]
remove LB[idx]
}
```

4.1. Tracking Results

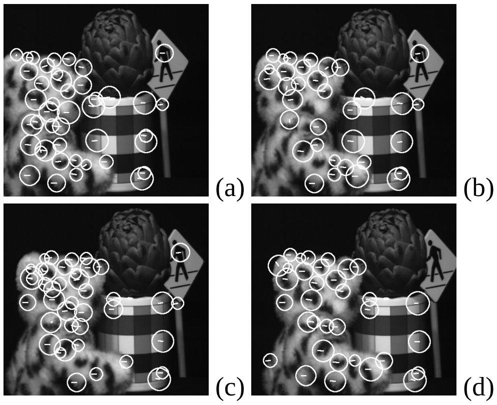


Figure 3. Frames from the CMU-VASC 'Artichoke' sequence with motion vectors and tracked salient regions calculated by our tracking algorithm overlaid

We present the results of our tracking algorithm when applied to two different image sequences. It should be stressed that the current tracker implementation only tracks salient regions between pairs of image frames, and doesn't incorporate any form of path coherence function or motion constraints [11]. As with the matching algorithm implementation, no kind of geometric consistency checks are carried out between the individual salient regions. The first sequence is the 'Artichoke' sequence from the CMU-VASC Image Database [1]. The second sequence shows a number of salient features that exhibit a scale change.

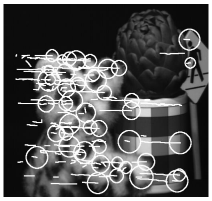


Figure 4. The overall path generated by the algorithm on the 'Artichoke' sequence

Four frames, with motion vectors showing the motion of the salient regions, from the 'Artichoke' sequence are shown in Figure 3. The overall path is shown in Figure 4. The results show that the algorithm was able to track the salient regions with reasonable accuracy (i.e. the magnitude and direction of the majority of the motion vectors was correct), although some mismatches occurred. The algorithm's performance is relatively poor when compared to other feature trackers, such as the Shi-Tomasi-Kanade tracker, or one of its extensions (e.g. [12]). However, most other feature trackers include motion constraints and incorporate some form of path-coherence function. Another advantage of our relatively simple tracking algorithm is that it does not involve any iterative computation.

The frames from the simple 3D tracking experiment are shown in Figure 5, with the motion-vectors and salient regions overlaid. A plot of the estimated trajectory of the salient features are shown in Figure 6. It should be noted that this experiment was performed in near real-time, with a frame rate of about 4 FPS and a latency of under one second, on a machine with a live Firewire (IEEE1394) camera and dual 867MHz PowerPC 7455 (G4+) processors (the processing is only performed on one processor, however). The results show that the estimated trajectories closely follow the actual trajectories taken by the regions.

5. Future Work

We have a number of plans for future development of this work. Firstly we aim to improve the performance of the saliency algorithm with respect to noise in the input image. Secondly we plan to develop some sort of geometric consistency function for the matching process. This should en-

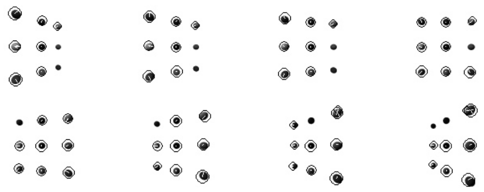


Figure 5. The frames used to demonstrate the 3D tracking functionality, overlaid with motion vectors

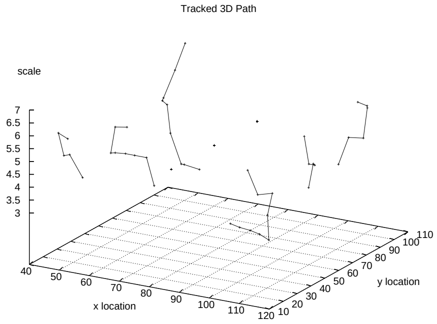


Figure 6. The 3D path of the salient region

sure that spatial arrangement of the matched salient points is consistent with the arrangement in the query image, subject to a range of geometric transforms. Some form of graph matching algorithm, such as [10], may be suitable for this process. Thirdly, we would like to improve the matching algorithm by using more powerful feature descriptors, such as colour-invariant histograms. We also would like to investigate whether the matching algorithm is suitable for content-based retrieval purposes, such as by matching through a user selected subset of salient features in the query image. Finally, we plan to build on our existing tracking algorithm implementation by incorporating motion constraints and path consistency.

6. Conclusions

In this paper, we have presented a novel technique for image matching and object tracking. Our method is based on the Scale-Saliency algorithm which we use to generate sparse representations of features within the image. These features are invariant to translation, rotation, and scaling. We demonstrate a technique for matching images (or sub-images) using the sparse feature representation. The technique may also provide a basis for content-based retrieval. We apply the matching algorithm to tracking features in a video sequence, by posing the feature-tracking problem as an image matching problem across pairs of video frames. Our tracking technique has the advantage that as a consequence of using the Scale-Saliency algorithm, it is possible to directly extract the 3D trajectory of salient regions between video frames.

7. Acknowledgements

We are grateful to the EPSRC and Motorola UK Research Laboratory for their support of this work.

References

[1] Carnigie Mellon University. Vision and autonomous systems center's image database. `http://vasc.ri.cmu.edu/idb/`.

[2] S. Gilles. *Robust Description and Matching of Images*. PhD thesis, University of Oxford, 1998.

[3] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Res.*, 40(10-12):1489–1506, 2000.

[4] L. Itti and C. Koch. Computational modelling of visual attention. *Nat. Rev. Neurosci.*, 2(3):194–203, 2001.

[5] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998.

[6] T. Kadir. *Scale, Saliency and Scene Description*. PhD thesis, University of Oxford, Deptartment of Engineering Science, Robotics Research Group, University of Oxford, Oxford, UK, 2001.

[7] T. Kadir and M. Brady. Saliency, scale and image description. *Int. J. Comput. Vis.*, 45(2):83–105, 2001.

[8] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum. Neurobiol.*, 4(4):219–227, 1985.

[9] N. Sebe, Q. Tian, E. Loupias, M. Lew, and T. Huang. Evaluation of salient point techniques. In *Image and Video Retrieval*, pages 367–377. Springer, July 2002.

[10] A. Shokoufandeh, I. Marsic, and S. Dickinson. View-based object recognition using saliency maps. *Image Vis. Comput.*, 17(5-6):445–460, 1999.

[11] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*, chapter 15, pages 696–708. PWS, second edition, 1999.

[12] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *1998 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1998.

[13] University of Washington. Ground truth image database. `http://www.cs.washington.edu/research/imagedatabase/groundtruth/`.