2005

Incorporating Background Invariance into Feature-Based Object Recognition

Andrew Stein
Carnegie Mellon University

Martial Hebert
Carnegie Mellon University, hebert@ri.cmu.edu

Published In
Seventh IEEE Workshop on Applications of Computer VIsion (WACV), .

Incorporating Background Invariance into Feature-Based Object Recognition

Andrew Stein[*] and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213
anstein@cmu.edu, hebert@ri.cmu.edu

Abstract

Current feature-based object recognition methods use information derived from local image patches. For robustness, features are engineered for invariance to various transformations, such as rotation, scaling, or affine warping. When patches overlap object boundaries, however, errors in both detection and matching will almost certainly occur due to inclusion of unwanted background pixels. This is common in real images, which often contain significant background clutter, objects which are not heavily textured, or objects which occupy a relatively small portion of the image. We suggest improvements to the popular Scale Invariant Feature Transform (SIFT) which incorporate local object boundary information. The resulting feature detection and descriptor creation processes are invariant to changes in background. We call this method the Background and Scale Invariant Feature Transform (BSIFT). We demonstrate BSIFT's superior performance in feature detection and matching on synthetic and natural images.

1. Introduction

Feature-based methods are commonly used for object recognition. Such approaches seek to efficiently match objects from a database to those seen in novel images using a sparse set of information-rich *features* extracted from images. Because the features only require local support, matching can be successful even in highly cluttered scenes. And when the features are designed to be extremely distinctive, only small numbers of matches are necessary to provide high confidence in an object's detection. These methods have also been motivated from a biological perspective [14, 24].

There are two broad steps involved in any feature-based scheme. First, features – also referred to as *keypoints* or *interest points* – are detected within an image. For this step, repeatability of detection is crucial [15]. If the detected locations of interest points on an object vary from image to image, there is no hope of successful matching. Second, signatures – or *descriptors* – are computed from local image values for each detected interest point in order to distinguish between them. The goal is to design a highly distinctive descriptor for each interest point to facilitate meaningful matches, while simultaneously ensuring that a given interest point will have the same descriptor regardless of the object's pose, the lighting in the environment, etc. So we see that both steps, detection and description, rely on invariance to various properties for success.

Much past work in this area has focused on producing interest points and descriptors that are invariant to scaling of the image [14, 15, 11, 7]. All of these approaches operate in scale-space to detect each interest point's characteristic scale [9, 8, 13]. Rotation invariance during *detection* is generally accomplished "for free" by using rotationally invariant image measures, such as the Laplacian. There are generally two approaches for creating rotationally invariant *descriptors*. Lowe [14] attaches a coordinate frame to each descriptor while others, e.g. [15], again use rotationally-invariant measures computed locally, such as local jets [8, 21].

Lowe's features have some invariance to affine transformation engineered into them, but others have designed truly affine-invariant features. In [2], Baumberg describes a method for creating affine-invariant descriptors around scale-invariant interest point detections. Mikolajczyk and Schmid go a step further in [16], combining the detection and description phases into an iterative scheme such that *both* steps are invariant to affine transformation.

Unfortunately, because all of these methods rely on the use of local image information at various scales, features whose descriptors overlap the object and the background will incorporate information from both. In fact, many features' detected *locations* will also be affected by the background. Therefore, when the object is seen with different backgrounds, its features would necessarily be different

[*] This research was made possible with support from the Air Force Office of Scientific Research.

(both in location and description). In particular, this problem is exacerbated as the size of the object relative to the image decreases. As small features completely contained within the object become impossible to detect for lack of resolution, we must depend on larger-scale features, which are more likely to overlap with the background. Ideally, the detection and description of an object's features would be the same regardless of the background on which it is seen. This implies that there is another type of invariance which would be desirable for feature-based methods: background invariance. We seek to address such a notion in this paper. We would like to incorporate object-background (often called figure-ground) separation into the detection and description processes in order to achieve such invariance. Knowledge of this separation could be obtained from various sources, including stereo disparity [3], motion cues [4], local or global segmentation schemes [26, 23], simple background subtraction, or a combination of these methods [19].

It should be noted that the discussion thus far has centered around features derived from image intensity information directly. Recently, edge-based features have emerged which exhibit some degree of background invariance in order to recognize wiry shapes in cluttered scenes [6, 17, 10]. Such methods build local features from edge maps in order to capture shape rather than texture. It is likely that future feature-based systems will leverage both shape and texture information for successful recognition of a wide variety of objects. To our knowledge, the work desribed in the remainder of this paper is the first to address background invariance when using texture information.

2. Background Invariant Detection

Our method builds directly on Lowe's Scale Invariant Feature Transform (SIFT) because of its popularity. Relevant portions of the SIFT method will be briefly reviewed here since our modifications occur at a fairly low level, but for complete details see [14]. Initial keypoints are detected as local extrema of a scale-space Laplacian pyramid. In practice, this Laplacian pyramid is efficiently approximated by the construction of a Difference-of-Gaussian pyramid instead. It is here, at the very beginning of the detection process, where we introduce our first modification.

Note that smoothing an image with a Gaussian filter blurs information across object-background boundaries. We can, however, replace this isotropic smoothing with more local process which respects arbitrary boundary conditions. This idea of using *anisotropic* smoothing has been extensively studied, e.g. [20, 25, 5]. We will only present here the basic concepts relevant to the remainder of the paper. The key idea we use is that Gaussian smoothing is equivalent to performing iterative local heat diffusion according to:

$$I^{(k+1)}(x,y) \leftarrow I^{(k)}(x,y) + \tau \nabla^2 I^{(k)}(x,y) \tag{1}$$

where,

$$\nabla^2 I(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{2}$$

That is, if we diffuse at each pixel (x, y) in an image I for k iterations according to (1), then as $\tau \rightarrow 0$, we have equivalently smoothed the image by a Gaussian with $\sigma = \sqrt{2k\tau}$. To keep the diffusion process numerically stable, we use $\tau = 0.2$. The advantage of using a local diffusion process is that we can naturally enforce arbitrary boundary conditions anywhere in the image, namely at object boundaries. In practice, we use Neumann boundary conditions when computing the second derivatives necessary for (2), allowing us to switch the diffusion process on or off according to local boundary information. While most prior work determines the amount of local diffusion based on local intensity gradient information, we use boundary information from an *outside* source. Otherwise, *every* edge in the image could suppress information flow across it, while we are only interested in preventing information flow across those edges that correspond to object boundaries. We leave the discovery of these edges to a separate process, which is an interesting problem in itself and is addressed further in the results section.

We can now build a boundary-respecting Gaussian pyramid by pausing our diffusion process at appropriate intervals to save each desired level. As pointed out by Lowe [14], if we start with an initial smoothing of σ_0, the desired scale-normalized Laplacian pyramid can be closely approximated by a Difference-of-Gaussian pyramid with the degree of smoothing at each level L chosen according to $\sigma_L = 2^{L/s}\sigma_0$, where s is the number of samples per scale octave (typically 3 or 4).

Once the Difference-of-Gaussian pyramid is constructed, sub-pixel local extrema are identified in scale-space and low-contrast and edge-like features are filtered out, all following [14]. A synthetic example is shown in Figure 1, where an image of a Sony Aibo[1] has been pasted onto two different textured backgrounds. Interest points are plotted as X's with circles representing their corresponding scales. Only those detections whose locations and scales are the same regardless of background are shown (with lines connecting them). In the center we see that when using regular SIFT, with no boundary information, only 38.3% of the detections with the brick background are also detected at the same location with the rocks background. On the right, where boundary information (derived from the known silhouette of the Aibo) was incorporated into the detection process as described above, 98.4% of the

[1] *Aibo* is a registered trademark of Sony Corporation.

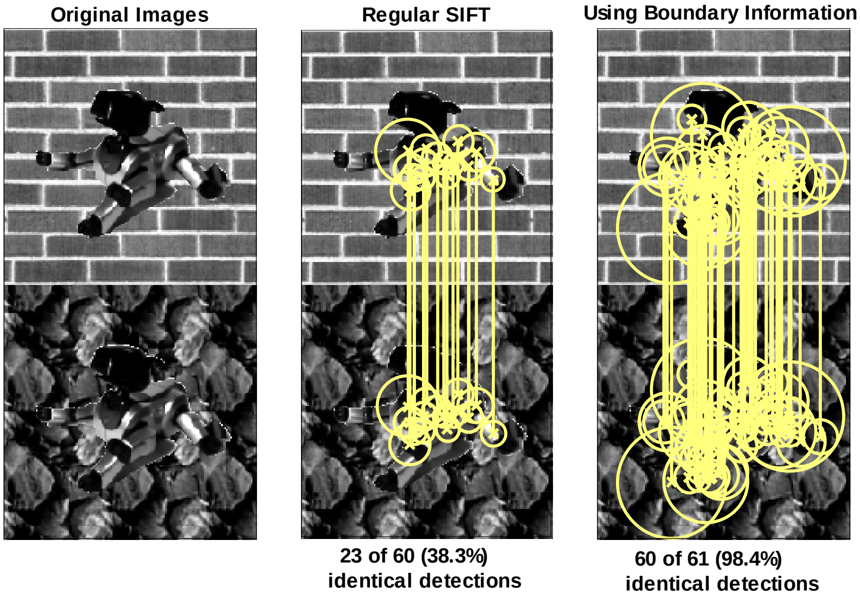

Figure 1. The incorporation of boundary information allows for more consistent feature detection on the same Sony Aibo object image pasted onto different background images.

detections are the same regardless of background. We emphasize that correspondence here is only in the sense of location and scale. No notion of descriptor matching has been incorporated into this example. Note how using boundary information allowed for consistent detection of the larger features, which are arguably more meaningful [24], as well as detection of more features along the narrow legs of the Aibo.

3. Background Invariant Description

Now that we have a method for detecting interest points in a background-invariant manner, we need to generate descriptors for those interest points which will also remain the same regardless of background. Again, our descriptors are constructed much like Lowe's SIFT descriptors, so for further details see [14].

For each interest point, image gradient magnitudes and orientations are extracted within a patch whose size is determined by the scale of the interest point. The orientations are put into a histogram weighted by their magnitudes and by distance from the patch center in order to determine a dominant orientation for each interest point. (Note that, in practice, multiple peaks in this orientation histogram may result in multiple interest points at the same location, but with differing orientations.) At this point, without any modification, background invariance is violated since we are working with a patch of image values which will often overlap object and background pixels. In order to impose background invariance on this step of the descriptor generation, we use a boundary-respecting weighting mask in the creation of the orientation histogram, rather than a simple Gaussian. This lessens the importance of samples further from the interest point while also effectively removing samples not on the object.

The creation of a weighting mask by heat diffusion as above would seem to be the natural choice here, but it turns out not to have the desired effect in this case. Figure 2 compares different weighting masks for an interest point marked by the red X with local boundary information as shown in (a). Without utilizing the local boundary information, a simple Gaussian weighting mask (b) as used by SIFT will clearly give non-zero weight to pixels which do not lie on the object, resulting in a non-background-invariant descriptor. But if we diffuse outward from the interest point according to (1) in order to produce a boundary-respecting Gaussian mask (c), we see that the weight values build up like a "snowdrift" at the boundaries, resulting in a weight mask which is no longer truly a function of distance from the interest point (note that the darkest part of the mask is not at the X as we would like). If we use a distance marching procedure [22] to propagate distances away from the interest point while respecting boundary information, however, we obtain the correct weight mask (d). Such a mask gives zero weight to pixels not on the object and weight proportional to distance from the interest point for pixels on the object. In addition, this procedure is more efficient than repeated diffusion at every interest point.

Finally, we also found that adding a weighting mask based on distance from the nearest boundary increased per-

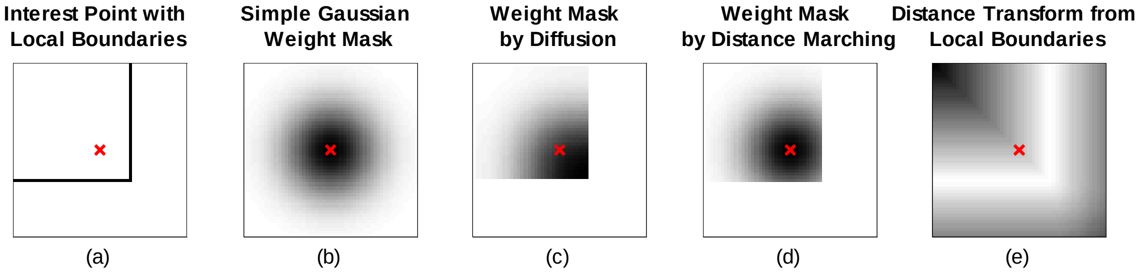
Interest Point with Local Boundaries
Simple Gaussian Weight Mask
Weight Mask by Diffusion
Weight Mask by Distance Marching
Distance Transform from Local Boundaries
(a)
(b)
(c)
(d)
(e)

Figure 2. Various weighting masks around an interest point for incorporating a nearby object boundary into descriptor creation.

formance. This weighting function can be computed using a distance transform from local boundaries and is pointwise multiplied by the Gaussian weighting mask described above. An example is given in Figure 2 (e). The idea here is to lessen the contribution of pixels which lie close to object boundaries, whose precise locations are not generally known in practice. Thus the final weighting mask becomes a balance between increasing weight as we approach the interest point and decreasing weight as we approach a nearby boundary.

Once a dominant orientation is assigned, a coordinate system is aligned to that direction. The gradient orientations at each site within the local patch are placed into sixteen histograms, each with eight bins, positioned relative to the aligned coordinate system. Each orientation sample's contribution to a given histogram is weighted by its magnitude, distance from that histogram's position in the coordinate frame, and distance from the interest point itself. This process is illustrated in Figure 3. For the interest point marked by the green X at the center, a coordinate frame has been aligned to the dominant gradient orientation as found by the procedure outlined above. Each square in the grid represents a sample site for the patch surrounding the interest point, where a sample consists of the gradient orientation and magnitude for that location in the image. The sixteen histogram centers are designated by the red dots. The blue shaded sample is shown to contribute to its four nearest histograms with a weight which is bilinearly interpolated based on distances to those histograms' centers.

To add background invariance to this standard SIFT descriptor, a boundary-respecting weighting mask produced by fast marching is again used to weight each sample's histogram contribution according to its distance from the interest point. The distance-from-nearest-boundary weight as described above is used here as well. Finally, all sixteen histograms' bin values are concatenated into a 128-vector, which is then normalized into a unit vector. Following [14], this vector is further adjusted by clipping all values at 0.2 and then renormalizing. The resulting vector forms the 128-dimensional descriptor for an interest point. In Figure 4, we have artificially chosen the same interest point for the Sony Aibo object, but with different backgrounds. Note that the large scale of the interest point and the shape of the object cause significant inclusion of background pixels. We see in (c) that the SIFT descriptors computed with the brick background (upper bars) and the rocks background (lower bars) are very different. There is no hope of the two being matched as the same descriptor. But when boundary information is incorporated into the weight masks as described, we see that the resulting descriptors (d), are exactly the same.

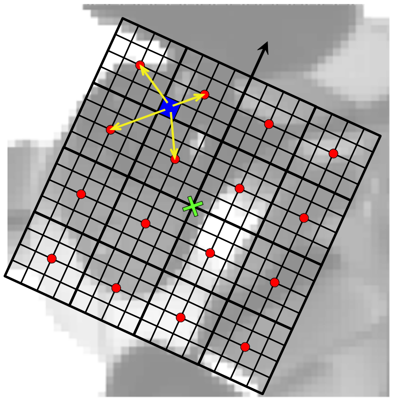

Figure 3. Descriptor creation from a set of gradient orientation histograms aligned to an interest point.

We now have all the pieces necessary for what we refer to as a Background and Scale Invariant Feature Transform (BSIFT). In the following sections, we will compare the performance of our BSIFT method to standard SIFT.

4. Synthetic Results

In order to evaluate our method's performance, we need to know ground truth object boundaries. To facilitate large-scale evaluation with a database of objects and a non-

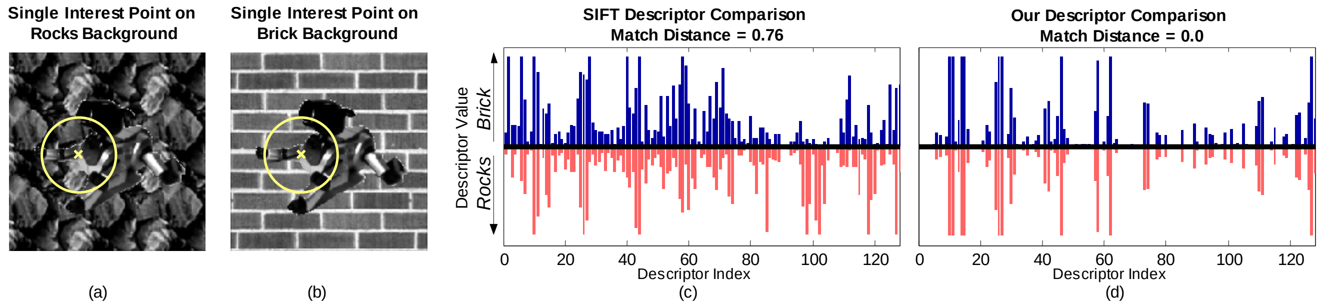

Figure 4. Incorporating boundary information into the descriptor creation allows us to compute exactly the same descriptor for an interest point, despite its overlap with two different backgrounds. SIFT, on the other hand, produces drastically different descriptors.

trivial number of example images, we artificially paste object images from a database of 110 objects onto a set of 25 background images. The set of backgrounds consists of about half indoor office/lab scenes and half outdoor natural scenes. The database of objects and backgrounds is a concatenation of some of our own images with many others from various online databases [18, 1, 12]. Each object has been hand segmented into foreground and background pixels, so precise object boundaries are known. Furthermore, features have been extracted from the objects with and without using boundary information (i.e. using BSIFT and SIFT, respectively), and all of these features are stored in one large database consisting of over 8000 features for each method. In order to guard against implementation differences and ensure the only variable being tested is the inclusion or lack of boundary information, we do not use Lowe's publicly-available SIFT library to test against our BSIFT implementation. Rather, we use only our own code, supplying boundary information to test BSIFT or witholding boundary information to test SIFT. (Without any boundary information, our method defaults to using standard Gaussian smoothing and Gaussian weighting masks.) We therefore expect that our baseline performance will differ from the more mature SIFT library available from Lowe. Finally, it is worth noting that the database includes many low-texture objects as well as objects with narrow appendages. Both are difficult for existing feature-based methods. A random selection of some of the objects in our database is shown in Figure 5. The database is also available online at http://www.cs.cmu.edu/˜stein/BSIFT.

For each test, a random background and object pair is chosen and the object is pasted onto the scene at a random location, orientation, and scaling (60-100%). In addition, we can simulate to some degree the effect of errors in object boundary extraction by introducing random contiguous breaks in the boundaries. This degradation is quantified by the percentage of the original boundary that is eroded away. Our performance criterion is the percentage of interest points present for the object in the database that are correctly matched to the object in the generated image - where a correct match implies that both the object's identification and location are correct. For reliable results, [14] suggests defining a match to be found when the ratio between the closest and second-closest descriptors in the database, using simple Euclidean distance, is less than a threshold (currently 0.6 in for the tests described here). The interesting cases are those when our method finds substantially more matches than SIFT, or better still, when SIFT fails to find any matches and our method succeeds.

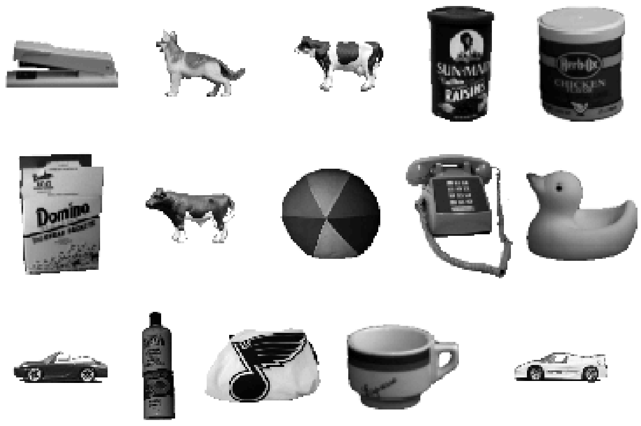

Figure 5. A few examples from the 110-object database used in our synthetic experiments.

For a set of 3000 experiments, we found that the average correct match (true positive) rate using our method was 80.1% while SIFT's was only 55.4%. Our method had a strictly greater number of correct matches (meaning more confident object recognition) 68.1% of the time. In fact, our method found at least one match while SIFT failed

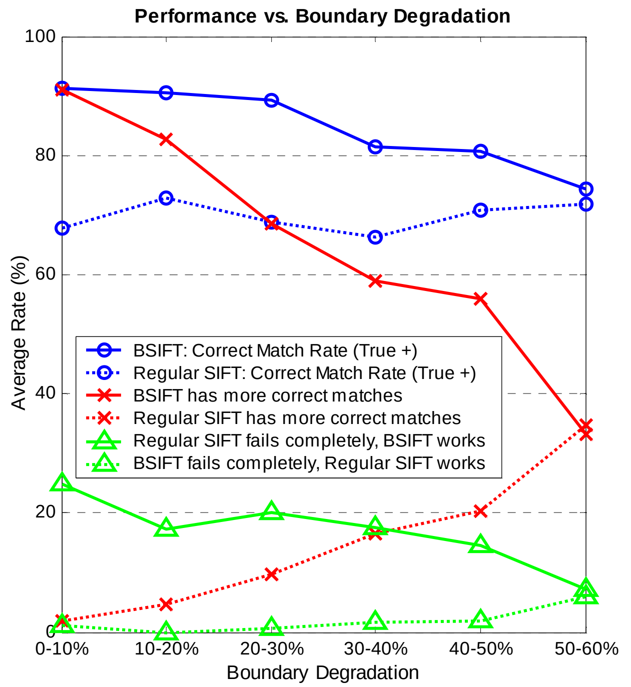

Figure 6. As expected, as boundary degradation increases, BSIFT's performance resembles that of regular SIFT more and more.

completely in 26.3% of the experiments. In 9.3% of the experiments SIFT found more matches than our method, and in 15.0% neither method found any matches (thus demonstrating the existence of some rather difficult objects in our database).

As expected, our method's performance and its superiority to regular SIFT decrease as boundary degradation increases since information derived from object boundary locations is the only difference between the two methods. Figure 6 shows a plot of performance values versus the amount of boundary degradation for a set of 1000 experiments in which scaling was set at 100% and boundary degradation varied between 0 and 60%. Note how BSIFT's performance falls off gracefully rather than completely failing as boundaries are eroded away and information leaks through during the diffusion and distance marching processes. Finally, in Figure 7, we also see that BSIFT is more resistant to the effects of scaling. For a set of 1000 experiments in which scaling varied uniformly between 60 and 100% (while edge degradation was disabled), we see that the matching rate for BSIFT does not fall off as significantly as for SIFT. For BSIFT the matching rate drops about 10% as the scaling decreases, whereas SIFT's performance – which is consistently much lower than our method's – drops more than 20%.

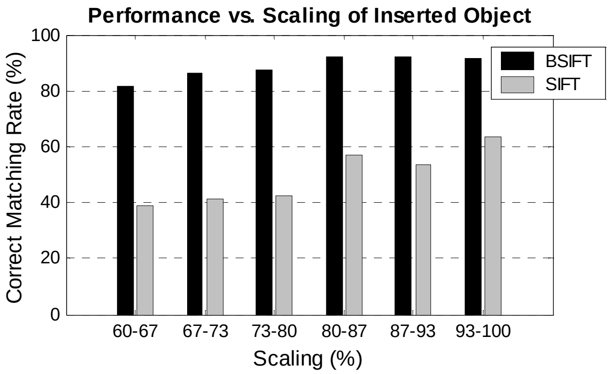

Figure 7. As scaling decreases, SIFT's performance remains below that of BSIFT and falls off more quickly. BSIFT achieves over 80% correct matching performance over the scaling range.

5. Results from Real Data

As suggested above, some possibilities for obtaining the required boundary information necessary for BSIFT include stereo disparity, motion parallax, various local or global segmentation schemes (e.g. graph cuts), or simple background subtraction. We present results using background subtraction and stereo disparity to demonstrate our method's viability in practice. For the following examples, we emphasize that the small number of matches (relative to typically reported SIFT results) is due to extremely limited resolution and fairly low-texture objects. These cases are not often considered elsewhere, but are important in practice.

In the top row of Figure 8, a toy car has been placed on a desktop such that it is quite small with respect to the image size. Thus, there is not enough resolution to find many features on the object. We see that SIFT finds two feature matches while BSIFT finds six. In the bottom row, a set of dry-erase markers – which exhibit very little texture – have been placed in an office scene, again such that they are small with respect to the image size. Here, SIFT fails to find any matches, while our method finds four. Note that there are only nine and eleven total features detected for the markers' *training* image using SIFT and BSIFT, respectively.

Figure 9 shows results using boundary information derived from a stereo disparity map. In this case the test image has been downsampled to 160 × 120, while using the full-resolution training image. The smaller features normally available on the fan, which regular SIFT could correctly match, are of little use in this case because of the lack of resolution. The texture of the couch in the background contaminates any largescale SIFT features, but BSIFT is

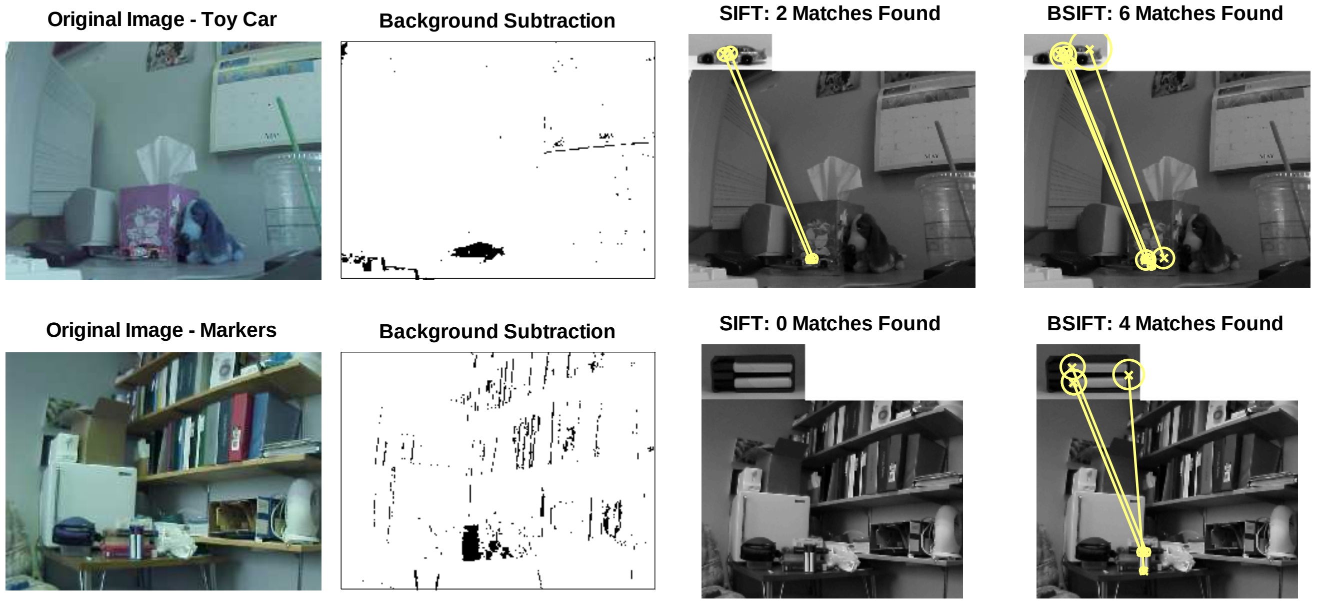

Figure 8. BSIFT outperforming SIFT using background subtraction for a toy car object (top row) and a set of dry-erase markers (bottom row). Note that only three matches for the dry-erase markers are visible because two are at the same location but with different orientations.

able to correctly match them. Indeed, SIFT fails to find a single match, while BSIFT finds four.

6. Conclusion

In this work, we have explicitly introduced the notion of background invariance as an additional type of invariance necessary for feature-based object recognition methods to perform well in general applications. We have developed a straightforward set of modifications based on heat diffusion and distance marching for introducing background invariance to the detection and description processes of the popular SIFT algorithm. We have also shown on fairly large-scale synthetic tests as well as some real experiments that the use of object boundary knowledge can improve feature matching performance significantly. We reiterate that while we have outlined a SIFT-based approach, the addition of background invariance will likely improve *any* feature-based method. SIFT, however, is a natural choice due to its popularity in the literature and in application.

With the basic principles and methods defined, the main thrust of future research will focus on robust extraction of object boundaries to use with our BSIFT approach. We will also investigate our method's sensitivity to boundary localization.

In its current form, however, this work provides for the first time a foundation for incorporating boundary information into intensity feature detection and description as well as results indicating the approach's practical utility.

Acknowledgements

We would like to thank David Lowe, Bob Collins, Yan Ke, and Dave Tolliver for their helpful discussions and Tom Stepleton for providing some of the object and background images used in our experiments.

References

[1] K. Barnard, L. Martin, B. Funt, and A. Coath. A data set for colour research. *Color Research and Application*, 27:147–151, 2002.

[2] A. Baumberg. Reliable feature matching across widely separated views. In *CVPR*, pages 774–781, 2000.

[3] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, 1998.

[4] M. Black and D. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, 2000.

[5] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heegar. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7:421–432, 1998.

[6] O. Carmichael and M. Hebert. Shape-based recognition of wiry objects. In *CVPR*, June 2003.

[7] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *CVPR*, pages 612–618, June 2000.

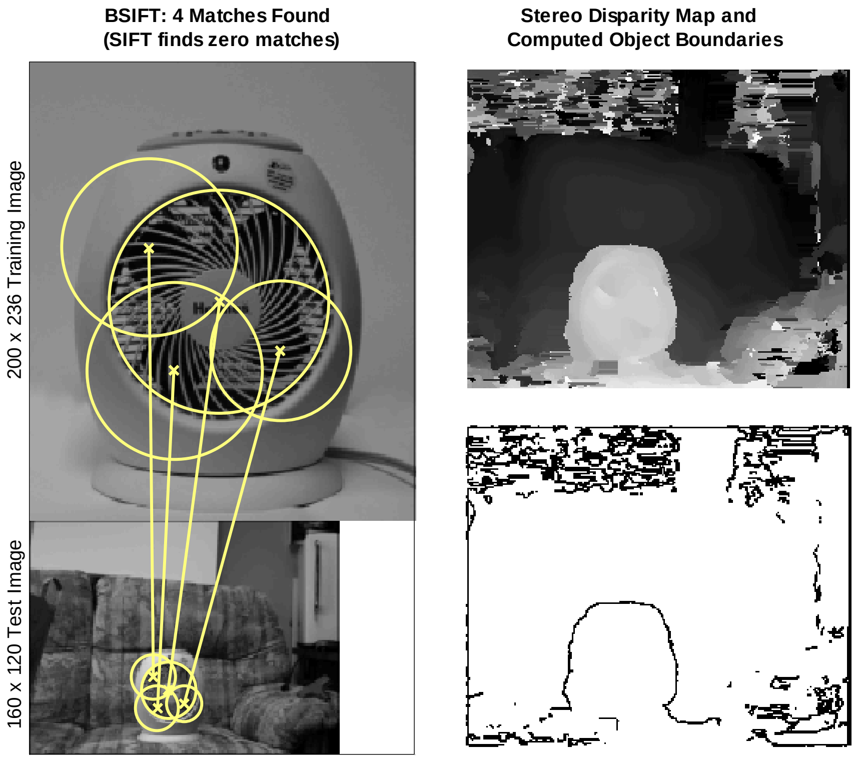

Figure 9. Using object boundary information derived from a stereo disparity map, BSIFT is able to find matches on the fan despite the low resolution/scale. SIFT fails completely on this example. Note that the training (top) and test image (bottom) are shown at actual relative scale.

[8] L. Florack, B. Romeny, M. Viergever, and J. Koenderink. The gaussian scale-space paradigm and the multiscale local jet. *IJCV*, 18:61–75, 1996.

[9] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388, 1992.

[10] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, 2004.

[11] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2):83–105, 2001.

[12] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR*, June 2003.

[13] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure. In *ECCV*, pages 389–400, May 1994. Stockholm, Sweden.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, January 2004.

[15] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, pages 525–531, 2001.

[16] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, pages 128–142, 2002.

[17] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. *BMVC*, 2003.

[18] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, February 1996.

[19] P. Nordlund. *Figure-Ground Segmentation Using Multiple Cues*. PhD thesis, University of Stockholm, May 1998.

[20] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, July 1990.

[21] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5):530–534, May 1997.

[22] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, 1997.

[24] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, July 2002.

[25] Y. L. You, W. Xu, A. Tannenbaum, and M. Kaveh. Behavior analysis of anisotropic diffusion in image processing. *IEEE Transactions on Image Processing*, 5:1539–1553, 1996.

[26] S. Yu and J. Shi. Object-specific figure-ground segregation. In *CVPR*, 2003.