Copyright 2013 IEEE. Published in the IEEE 2013 International Conference on Image Processing (ICIP 2013), scheduled for September 15 - 18, 2013 in Melbourne, Australia. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

FAST CORTICAL KEYPOINTS FOR REAL-TIME OBJECT RECOGNITION

Kasim Terzić   *J.M.F. Rodrigues*   *J.M.H. du Buf*

Vision Lab (LARSyS), FCT, University of the Algarve, Gambelas Campus, 8000 Faro, Portugal

ABSTRACT

Best-performing object recognition algorithms employ a large number features extracted on a dense grid, so they are too slow for real-time and active vision. In this paper we present a fast cortical keypoint detector for extracting meaningful points from images. It is competitive with state-of-the-art detectors and particularly well-suited for tasks such as object recognition. We show that by using these points we can achieve state-of-the-art categorization results in a fraction of the time required by competing algorithms.

***Index Terms*—** Computer vision, Object recognition, Image classification, Gabor filters, Real time systems

1. INTRODUCTION

It is widely accepted that attention plays a central role in human vision and is partly responsible for the speed and reliability of our object recognition. Consequently, detection of interest points has attracted much attention from the computer vision community and there exist a number of approaches for detecting interest points which are stable under a wide range of transformations. They range from very general features [8, 23] to highly object-specific ones [13, 1], and everything inbetween [14, 16, 5]. Although early vision in humans has been studied in great detail and several influential object recognition methods are biologically motivated [4, 22, 17], biologically inspired keypoints are rarely used in computer vision, usually due to excessive computation times [20] or lack of multi-scale analysis [7, 9].

Given the number of interest point detectors, it is surprising that many of the best-performing object categorization algorithms do not employ such detectors, but process entire images instead, either by pre-processing entire images to obtain feature vectors [19], by sampling descriptors on a dense grid [2, 25, 19] or by processing entire images hierarchically and detecting salient features in the process [4, 19, 22, 17]. These approaches provide a lot of data which helps classification, but also introduces a lot of redundancy, leading to either a long machine learning stage [25, 29, 26] or long recognition times [2]. Many influential methods which start by extracting interest points are no longer competitive [18, 12, 3]. Processing a large dataset like Caltech 101 [10] typically takes hours or days to complete. With the ever-increasing trend toward active and real-time vision, the need to quickly learn and recognize new classes of objects is making the question of data selection relevant again.

In this paper, we show that obtaining state-of-the-art categorization performance need not be slow. We introduce an improved biological keypoint detector which compares favorably to the state of the art. We then combine detected keypoints with a nearest-neighbor classifier and show that the combination can achieve state-of-the-art categorization performance which is orders of magnitude faster than previous algorithms.

2. CORTICALLY-INSPIRED KEYPOINTS

Our keypoints are based on the human visual cortex [20]. Each layer of cells is modeled as a filtering operation, each kernel corresponding to a typical weight profile of a particular type of cortical cell: simple cells, complex cells, end-stopped cells and inhibition cells (see Fig. 1). Simple cells are modeled using complex Gabor filters with phases in quadrature

$$g_{\lambda,\sigma,\theta,\phi}(x,y) = \exp\left(-\frac{\tilde{x}^2+\gamma\tilde{y}^2}{2\sigma^2}\right)\exp\left(i\frac{2\pi\tilde{x}}{\lambda}\right), \quad (1)$$

with $\tilde{x} = x\cos\theta + y\sin\theta$, $\tilde{y} = y\cos\theta - x\sin\theta$ and $\gamma = 0.5$. λ is the wavelength (in pixels), σ the receptive field size (also in pixels), which are related by $\sigma/\lambda = 0.56$, and θ determines the filter orientation (typically 8 orientations are used). Simple cell responses are obtained by convolving the image with the complex Gabor filter: $R_{\lambda,\theta} = I * g_{\lambda,\theta}$. Complex cells are the modulus of simple cell responses: $C_{\lambda,\theta} = |R_{\lambda,\theta}|$. Simple cells respond differently to line or edge stimuli, complex cells respond to both. Remaining kernels are sums of Dirac functions. If $ds = 0.6\lambda\sin\theta$ and $dc = 0.6\lambda\cos\theta$, double-stopped cell kernels are defined by

$$k^D_{\lambda,\theta} = \delta(x,y) - \frac{\delta(x–2ds, y+2dc) + \delta(x+2ds, y–2dc)}{2} \quad (2)$$

and tangential and radial inhibition kernels by:

$$k^{TI}_{\lambda,\theta} = -2\delta(x,y) + \delta(x+dc, y+ds) + \delta(x-dc, y-ds) \quad (3)$$

$$k^{RI}_{\lambda,\theta} = \delta(x+dc/2, y+ds/2) + \delta(x-dc/2, y-ds/2). \quad (4)$$

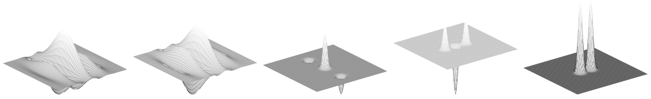

Fig. 1. Biologically-inspired filter kernels. From left to right: even simple cell, odd simple cell, double-stopped cell, tangential inhibition cell, radial inhibition cell.

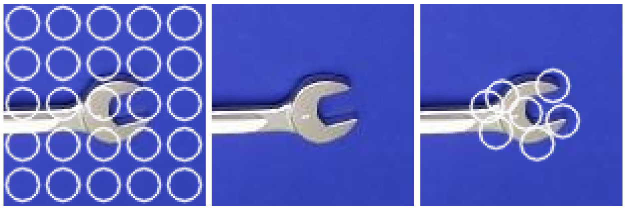

Fig. 2. Data selection step. Center: image from Caltech 101. Left: keypoints sampled from a regular grid. Right: our keypoints. Only one scale is shown for clarity.

The final keypoint map is obtained by combining the results:

$$K_\lambda^D = \sum_{\theta=0}^{\pi} \left| C_{\lambda,\theta} * k^D_{\lambda,\theta} \right|^+ - \sum_{\theta=0}^{2\pi} \left| C_{\lambda,\theta} * k^{TI}_{\lambda,\theta} + C_{\lambda,\theta^\perp} * k^{RI}_{\lambda,\theta} - C_{\lambda,\theta} \right|^+ , \qquad (5)$$

where $\theta^\perp$ is orthogonal to θ and $|.|^+$ represents suppression of negative values. After a thresholding step, local maxima in this map correspond to interest points. Events at different scales are detected by varying the wavelength λ of the Gabor functions which model simple cells.

The original approach from [20] is slow due to many convolutions and the results are not competitive with the state of the art (see Fig. 4). We therefore improved the original approach in several ways: (1) we use a Gaussian pyramid, limiting λ to $[4 \ldots 8)$. This speeds up the computation and eliminates instabilities at coarse scales caused by very large kernels in [20]. (2) We fit two 1-D parabolas around the maximum to obtain a more accurate keypoint position. (3) Since λ is small, transfer functions of Gabor wavelets are well-defined and we perform filtering in the frequency domain. (4) We only use double-stopped cells, which are more stable in practice.

These improvements make the algorithm practical by reducing runtime by several orders of magnitude. At the same time, they boost performance in standardized repeatability tests resulting in a state-of-the-art detector (see Fig. 4). Our CPU implementation takes a fraction of a second on a typical image, the GPU implementation easily runs in real time.[1] Figure 2 shows the difference between our keypoints and the grid sampling method.

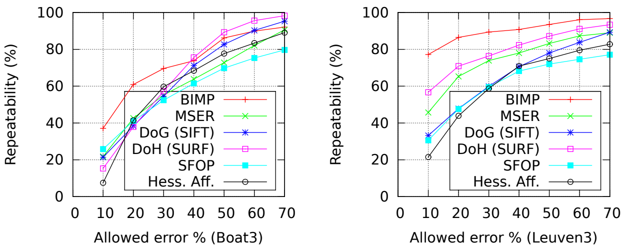

Fig. 3. Keypoint repeatability as a function of maximum allowed overlap error. Our algorithm is one of the best performing detectors.

2.1. Repeatability

We compared our keypoints with several state-of-the-art scale invariant detectors on the well-known repeatability benchmark by Mikolajczyk et al. [16]. We tested against SIFT [13], SURF [1] and the original biological operator by Rodrigues et al. [20], as well as two affine-invariant region detectors: Hessian affine [16] and MSER [14]. We used the original Matlab/C++ benchmark and author-supplied implementations of the Hessian affine, MSER and biological detectors, applying default parameters. For SIFT and SURF we relied on widely used OpenCV implementations. For convenience, we refer to our algorithm as "BIMP" (Biologically Inspired Multiscale keyPoints). We used seven scales, $\lambda \in \{8, 8\sqrt{2}, 16, 16\sqrt{2}, 32, 32\sqrt{2}, 64\}$, and eight orientations θ equally spaced on $[0, \pi)$. Figure 4 shows a selection of results. It can be seen that our detector performs very well, showing best-in-class performance in most cases.

In order to measure localisation precision, we have also plotted repeatability as a function of overlap error, as suggested in [5]. The results can be seen in Fig. 3: repeatability between the first and the third images of the Boat and Leuven sequences, respectively. It can be seen that our detector is very accurate compared to competing detectors. Table 1 shows the times our test computer (quad-core Intel i5, GeForce GTX 560 Ti) needed to process the first image of the Graffiti set, averaged over ten runs. We note the speedup over the original approach from [20], which shows that biologically inspired methods can be fast and practical.

3. OBJECT CATEGORIZATION

Our goal is to speed up recognition and make it feasible in real-time scenarios. Commonly used grid sampling yields large and redundant feature vectors, leading to long learning and/or classification times, but interest points have not been able to offer comparable results so far. In this paper, we apply the state-of-the-art *Local Naive Bayes Nearest Neighbor* classifier [15] and show that by using our keypoints for feature selection, we can achieve performance similar to the grid-based approach with only a fraction of the descriptors.

[1] The source code is available at `w3.ualg.pt/~kterzic/software.html`

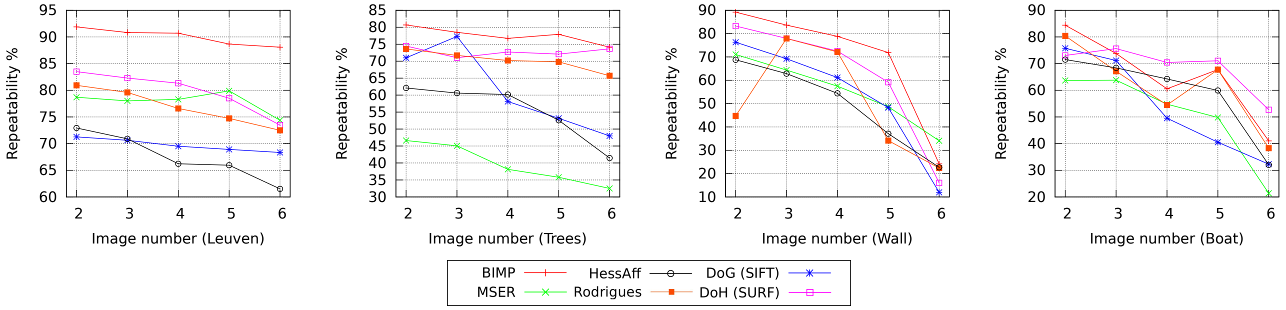

Fig. 4. Keypoint repeatability on standard image sets from Mikolajczyk *et al.* [16] Keypoints detected in the first image of a set are compared to those in images 2-6. Our method presents a notable improvement over the original method of Rodrigues *et al.* [20] and outperforms the state of the art in terms of repeatability in many cases.

Table 1. Runtime on the first image from the Graffiti set, 800×640 pixels (lower is better). Detectors prefixed with "G" are GPU-based implementations. GPU times were obtained using the OpenCV implementation and exclude GPU transfer times.

detector	Rodr. [20]	SFOP [5]	HesAff [16]	SIFT [13]	**BIMP**	SURF [1]	G-SURF	**G-BIMP**	G-ORB [21]
time (s)	3200	9.14	1.01	0.50	**0.32**	0.18	0.036	**0.028**	0.024

3.1. Categorization Approach

We extract keypoints at several scales and extract a SIFT descriptor at each keypoint location, with its size proportional to the keypoint wavelength λ. Scaling is chosen so descriptors are between 32 and 128 pixels large. Essentially this means dense sampling around points of high complexity and sparse sampling elsewhere. We then apply the local NBNN algorithm of [15], which we briefly summarize here for completeness. Each descriptor is augmented by its location in the image (we use the image center as the origin) scaled by a factor $\alpha = 2$. The conditional probability of a descriptor d_i given a class C is approximated by using r nearest neighbors:

$$p(d_i|C) \approx \frac{1}{L}\sum_{j=1}^{r} K(d_i - d_j^C), \tag{6}$$

where L is the total number of descriptors associated with class C in the training set, and K is the Parzen kernel (we use a Gaussian). Classification rule is given as a sum of log-odds:

$$C = \underset{C}{\operatorname{argmax}} \left[\sum_{i=1}^{N} \log \frac{P(d_i|C)}{P(d_i|\bar{C})} + \log \frac{P(C)}{P(\bar{C})} \right] \tag{7}$$

where $\bar{C}$ is the set of all classes other than C. $P(d_i|\bar{C})$ is approximated by a single sample, the nearest neighbor $r+1$. A set of K-D trees is constructed and searched in parallel for efficient nearest neighbor lookups.

When using a dense grid as in [15, 2], a single nearest neighbor per class is enough because of dense sampling of the feature space. In our case very few samples are used for probability density estimation, so more neighbors are needed: we use 60 nearest neighbors instead of the 10 used in [15].

3.2. Categorization Complexity

The complexity of the local NBNN classification algorithm is $O(cN_D \log(N_C N_T N_D))$, where N_C is the number of classes, N_T the number of training images per class, N_D the average number of descriptors per image, and c the number of times a K-D tree should be traversed. c, N_C and N_T are the same in our case, but a decrease in N_D (for example, by using our keypoints) results in a more-than-linear speed improvement. Lowering N_D from 2000 to 200 results in a speed-up of more than 12× on Caltech 101, and reduces memory requirements by a factor of 10.

3.3. Evaluation

We evaluate categorization performance on the Caltech 101 dataset [10] following standard evaluation procedures. We perform 10 random splits of the data into disjoint training and testing sets and report the mean classification rate and standard deviation. All images are scaled to a uniform size (long side scaled to $l = 300$px, preserving aspect ratio). Then features are extracted from the image at locations provided by our method and a dense grid for comparison. Since we are interested in feature selection and reducing the number of features needed for categorization, we report classification rates as a function of the number of used features. This is achieved by varying the number of scales and image size l. We are motivated by biological vision: coarse scale data propagates faster and leads to quicker (but less accurate) recognition.

Figure 5 shows the results. Our own keypoints consistently outperform the grid results when using a very low number of features. Good performance with very few features indicates that our keypoints consistently capture the most rel-

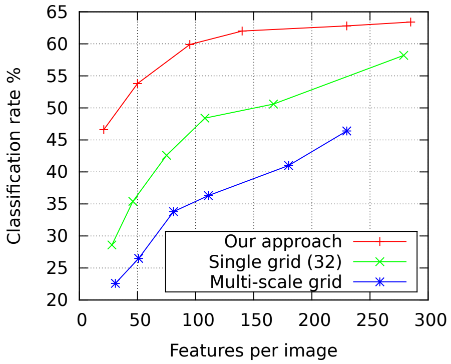

Fig. 5. Classification rate as a function of the number of used features in the case of 15 training images. The commonly used multi-scale grid (4 scales) does poorly with few features. Single-scale grid with a fixed feature size of 32 pixels does better due to a higher density of features. Our keypoint method performs consistently well and degrades gracefully.

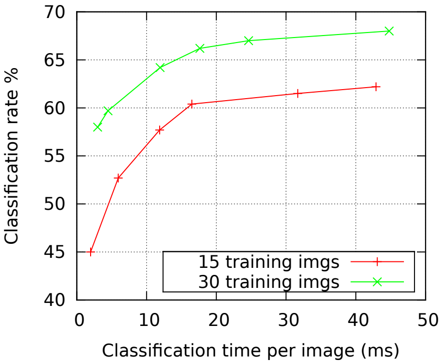

Fig. 6. Trade-off between classification time (in milliseconds) and accuracy on the Caltech 101 dataset. Slower recognition increases accuracy, but very good results can already be obtained at 50 frames per second. As a comparison, grid-based local NBNN needs about a second per image for the same performance, classic NBNN about 10 seconds [15]. LLC needs about 1/4 seconds per image but needs a long learning stage.

evant information in the image. As we increase the number of keypoints, our results asymptotically approach the best published results with a dense grid using an NBNN method. We note that in our experiments SIFT and SURF keypoints failed to match the grid-based approach.

Table 2 shows a comparison with state-of-the-art categorization methods using a single feature type. Our method can maintain state-of-the-art performance even when using only a fraction of the features used by the other methods. We emphasize that a dense grid provides a superset of our features and that we cannot expect to outperform these methods. However, our results show that good feature selection can achieve comparable results at a fraction of the cost (see Fig. 6). We note that with 30 training images we almost match the best reported NBNN result [15], using only 730 features per image! Since we are using the NBNN-based classifier from [15], only comparisons with other NBNN methods make sense, but we expect that combining our feature selection process with other classifiers such as LLC and ScSPM will enable a similar reduction in needed features for those algorithms.

4. CONCLUSIONS

While there has been much work on real-time object recognition for specific scenarios like robotics, most general-purpose categorization approaches dealing with hundreds of categories have sacrificed speed for accuracy. In this paper, we have addressed this problem by presenting a biologically-inspired method for data selection which significantly reduces the number of features needed for state-of-the-art accuracy. Feature extraction is fast on a CPU and runs in real-time on a GPU. The local NBNN classifier requires no learning and is fast if the number of features is small. The resulting system can be extended with new classes almost instantly, and recognize hundreds of classes at over 50 frames per second, making it optimal for active vision tasks such as robotics, where a speed-performance trade-off is inherent. This way a robot can choose between fast and approximate, and a slow and more accurate recognition.

Acknowledgements This work was supported by the EU under the FP-7 grant ICT-2009.2.1-270247 *NeuralDynamics* and the FCT under the grant PEst-OE/EEI/LA0009/2011 .

Table 2. Comparison to the state of the art in categorization. Results on Caltech 101 using SIFT descriptors. We only list results where the average number of features per image was reported by the authors. Numbers marked with (*) were calculated based on the reported sampling method (spacing and number of scales) and an average image size of 300x250 pixels. Note that [15] applies a contrast-based data selection step.

	# features	15 images	30 images
NBNN-based methods			
NBNN kernel [24]	2000	61.3±0.2	69.6±0.9
Our result	**140**	**62±1.3**	**67.2±1.2**
NBNN [24]	2000	62.7±0.5	65.5±1
Local NBNN [15]	1639	65	
Our result	**729**	**65.1±1.1**	**71.2±0.5**
Non-NBNN methods			
Gehler *et al.* [6]	3000*	54.5±0.9	63.8±1
SPM [11]	1172*	56.4	64.6±0.8
LLC [27]	3516*	65.4	73.4
ScSPM [28]	1400*	67±0.5	73.2±0.5
NBNN+phow [24]	2000	69.2±0.9	75.2±1.2

5. REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In *Proc. CVPR*, Anchorage, 2008.

[3] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71(3):273–303, 2007.

[4] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, Minneapolis, Jun 2007.

[5] W. Förstner, T. Dickscheid, and F. Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *ICCV*, Kyoto, Japan, 2009.

[6] P.V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009.

[7] T. Hansen and H. Neumann. Neural mechanisms for the robust representation of junctions. *Neural Computation*, 16:1013–1037, 2004.

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[9] F. Heitger, L. Rosenthaler, R. von der Heydt, E. Peterhans, and O. Kuebler. Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Res.*, 32(5):963–981, 1992.

[10] R. Fergus L. Fei-Fei and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Proc. CVPR Workshop on Generative-Model Based Vision*, Washington DC, 2004.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, New York, 2006.

[12] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, 2002.

[15] S. McCann and D.G. Lowe. Local naive bayes nearest neighbor for image classification. In *CVPR*, pages 3650–3656, Providence, 2012.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65:2005, 2005.

[17] J. Mutch and D. G. Lowe. Multiclass Object Recognition with Sparse, Localized Features. In *Proc. CVPR*, volume 1, pages 11–18, New York, 2006.

[18] B. Ommer and J. Buhmann. Learning the compositional nature of visual object categories for recognition. *IEEE T-PAMI*, 32(3):501–516, 2010.

[19] N. Pinto, D.D. Cox, and J.J. diCarlo. Why is Real-World Visual Object Recognition Hard? *PLOS computational biology*, 4(1):0151–0156, Jan 2008.

[20] J.M.F. Rodrigues and J.M.H. du Buf. Multi-scale keypoints in V1 and beyond: Object segregation, scale selection, saliency maps and face detection. *BioSystems*, 86:75–90, 2006.

[21] E. Rublee, V. Rabaud, K. Konolige, and G.R. Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, pages 2564–2571, Barcelona, Nov 2011.

[22] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Object recognition with cortex-like mechanisms. *IEEE T-PAMI*, 29(3):411–426, 2007.

[23] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593 – 600, 1994.

[24] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN kernel. In *ICCV*, Barcelona, Nov 2011.

[25] M. Varma and D. Ray. Learning The Discriminative Power-Invariance Trade-Off. In *Proc ICCV*, pages 1–8, Rio de Janeiro, 2007.

[26] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *Proc. CVPR*, pages 1597–1604, New York, 2006.

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, San Francisco, Jun 2010.

[28] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, Dec 2009.

[29] J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 73(2):213–238, Jun 2007.