Sliding Window Bag-of-Visual-Words for Low Computational Power Robotics Scene Matching

Daniel Ginn, Alexandre Mendes, Stephan Chalup, Zhiyong Chen
School of Electrical Engineering and Computing
Faculty of Engineering and Built Environment, University of Newcastle
Newcastle, Australia
e-mail: daniel.ginn@uon.edu.au, {alexander.mendes; stephan.chalup; zhiyong.chen}@newcastle.edu.au

***Abstract*—In this paper, we introduce a new method, based on a sliding window geometrical extension to Bag-of-Visual-Words (called swBOVW) intended for application to low computational power robots. Benchmarked against RANSAC as a geometric validator to BOVW, three implementations of this technique are presented to improve either the performance or the computational cost. The three implementations are: as a replacement to RANSAC as a geometric validator; as a supplement to RANSAC; and as a replacement to traditional BOVW when the number of images in the database can be reduced. Seeking to utilise some of the geometric information ignored by traditional BOVW, this technique is developed from the use of sub-regions in Spatial Pyramids, and applied to the matching of whole images. This technique is applied in the context of humanoid robotic soccer to the problem of field end symmetry, and provides geometric validation along the horizontal axis of images. When applied, the technique has been able to either halve the cases of unresolved image queries, or halve the computational cost required to achieve comparable results to the benchmark.**

Keywords-BOVW; bag-of-visual-words; image matching; robotics; machine vision

I. INTRODUCTION

The task of matching two scenes or images has attracted a great amount of research attention due to the plethora of applications, ranging from scene classification and facial recognition, all the way to Visual Simultaneous Localisation and Mapping (V-SLAM) techniques used in robotic navigation and augmented reality headsets. However, not all applications that could benefit from image matching have the computational resources available that current state-of-the-art methods require. Among the many techniques that have been able to achieve high levels of certainty, Bag-of-Visual-Words (BOVW) [1-2] is a particularly lightweight one - a trait of special interest for low computational power robotics. One primary hindrance however has been that the methods used to extract the features from images for BOVW, such as SIFT [3] and SURF [4], are computationally expensive. In addition, BOVW's exclusive reliance on frequency of feature occurrences in an image discards important geometric information. In this paper we are going to present a geometric BOVW extension called *sliding window BOVW* (swBOVW), inspired by the work done on Spatial Pyramids [5], with application to low computational power processors. In this paper, three alternative implementations of swBOVW are shown, either replacing the commonly used geometrical validation method of RANSAC [6] or complementing it, to provide a significant improvement to BOVW for computationally constrained robotic image matching applications. The specific application area is the RoboCup humanoid robotic soccer World Cup, where we focus on solving the field symmetry problem inherent in a soccer field with same colour goal posts. The remainder of this paper is organised as follows: section 2 gives a brief overview of related work and concepts; section 3 outlines the methodology used in this research; section 4 reports the results and improvements achieved; and section 5 closes with a conclusion and possible future research directions.

II. BACKGROUND

Bag-of-Visual-Words (BOVW) is a technique that allows for rapid comparisons between images. However, there are a few limitations associated to its use. The traditional techniques used to extract the features for the comparison, such as SIFT and SURF, are time-consuming, and the BOVW method does not use any information related to the spatial configuration of those features.

A. Related Work

A considerable amount of work continues to be done on enhancing BOVW to include geometrical information. These works include [7] who replaces RANSAC by augmenting the features used in BOVW image matching to include affine geometric information, resulting in the name change to Bag-of-Visual-Phrases. Similarly, [8] presents Region Similarity Arrangement, which uses polar coordinates of nearby points, using this to enhance the BOVW features. Reference [9] use image saliency to modify how the weighting of features is performed, along with using Nearest-Neighbour matching in a local region around features.

B. 1D SURF

In low-end, real-time computing, traditional feature extractors like SIFT and SURF are prohibitively computationally expensive. In response to this problem, Anderson et al. [10] proposed a one-dimensional variation of SURF that extracted a horizontal strip, a few pixels in height, from an image and used only changes in the horizontal direction to identify features. With these changes, he

978-1-5386-6338-7/18/$31.00 ©2018 IEEE

achieved a reduction in computational cost of three orders of magnitude. This change was intended for use on ground-based robots moving in a planar environment, so the reduction of the image to a strip parallel with the ground plane is justifiable.

The grey-scale row of pixels which is supplied to 1D SURF is obtained by taking the vertical sum of pixel intensities within the strip to compensate for any minor vertical deviations between images. Features are then located by sliding a filter along the strip to identify local intensity peaks/maxima. Different sized features are identified by scaling up the sliding filter. In traditional SURF, a detected local maxima must be greater than: (1) all the pixels surrounding it in both its own scale size, and (2) the scales immediately below and above it. The second condition is relaxed in 1D SURF so that a large number of low quality features is extracted, instead of a low number of high quality features. A feature descriptor is then created using the pixel intensities in a region surrounding the feature (see Fig. 1). For a detailed explanation about the 1D SURF method we refer the reader to [10].

C. *Bag-of-Visual-Words*

Bag-of-Visual-Words is a comparison technique originally designed for document matching, where word frequency was compared between documents. In its application to image matching, words are replaced with the features extracted through techniques such as SIFT and SURF. A database of images is first prepared offline, where these features are clustered into `terms', the equivalent of word roots. The frequency of these terms is then inverse weighted to emphasise distinctive terms, while minimising common ones. The term frequency for each of the database images is then represented as a vector. When a query image is provided, its term frequency vector is compared against each of the stored vectors to find their cosine similarity. Cosine similarity measures the cosine of the angle between two vectors and is calculated using the following formula [11].

$$CS(x,y) = \frac{x^T y}{||x||\ ||y||} \tag{1}$$

In our RoboCup implementation, the aim was to identify which side of the field the robot was looking at, i.e. away goal vs home goal. The top matching results (up to a maximum of eight) which are above a cosine score of 0.4 are then used in a voting system to determine which end of the field is present in the query image. Note that in our experiments the database consists of 18 images (9 for each end of the field). The voting system required that a minimum of three matches had been found, as well as a difference between away goal and home goal votes of at least two. If both of these conditions were not met, then a result of `unsure' would be returned.

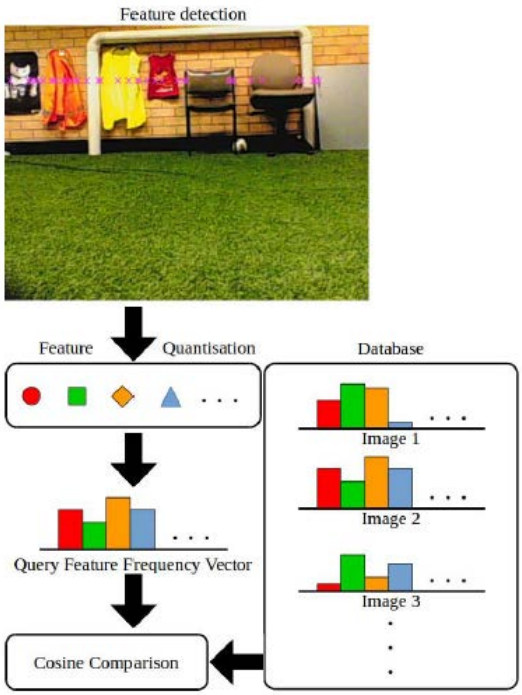

Figure 1. This figure provides an overview of how Bag-of-Visual-Words works. Features are first extracted from an image (location of the features is indicated with the crosses). They are then clustered into groups based on their feature descriptors, which is equivalent to grouping words which have a common root. Once reduced to a manageable number of feature groups, the occurrence frequency of each feature type is stored as part of a feature frequency vector. This vector is then compared with each vector in the database using a cosine similarity score. The motivation behind BOVW is that similar images will generate similar visual-word vectors, and thus their cosine similarity will be high. Figure modified from [2].

D. *Spatial Pyramids*

Spatial Pyramids is an extension to Bag-of-Visual-Words, with the purpose of retaining some of the spatial information that is discarded by BOVW. As features are being extracted from an image, their location is recorded. The image is then subdivided into quadrants, and BOVW is run again on each region (see Fig. 2). This technique has been used extensively for scene classification in the past. However in this paper the basic principles of spatial pyramids will be applied to match subregions of the query and database images.

E. *RANSAC*

RANSAC has been used by [12-13] to provide a spatial verification of the top BOVW results before the voting procedure took place, with [8] reporting that RANSAC continues to achieve state-of-the-art results. The feature locations of the query image and the database image being verified are placed on a XY scatter plot, with each axis representing the feature x-coordinates of one of the images.

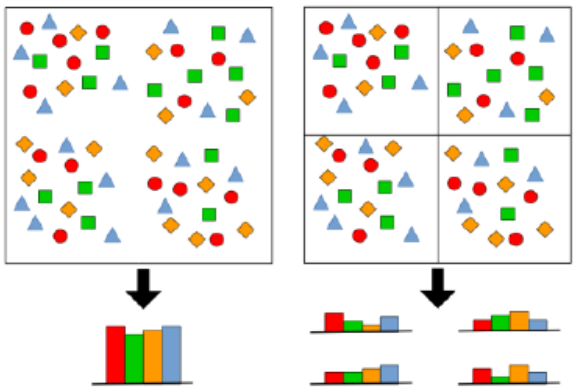

Figure 2. This figure provides the rationale for our own spatial Bag-of-Visual-Words extension. Here the Spatial Pyramid is described, simplified to 4 types of features - represented by circles, triangles, diamonds and squares. As the image is subdivided, information about the distribution of the features is recorded and can be used for more accurate matching. Note that each subdivision has its own feature frequency vector. Image modified from [5].

4		6	
9	8	2	1
2	7	3	10
A	B	C	D

(a) Query image

1.5	7.5		
9	6	4.5	
A	B	C	D

(b) Database image

Figure 3. (a) and (b) show the x-axis pixel location of the features in hypothetical query and database images respectively. The database image is exactly the same as the query image, but has been panned to the right by 3 pixels, as well as been scaled by a factor of 1.5, to replicate a common scenario. The database image has the same pixel width as the query image, so any features that fall outside the range of 1 to 10 are excluded. A, B, C and D represent 4 unique features. The resulting XY scatter plot of those features is shown in Figure 4.

Since each feature is not necessarily unique, each instance of a feature is plotted against all occurrences of that feature in the database image. A matching image will produce a high concentration of features near the diagonal from the origin (known as inliers). An example is provided in Fig. 3 and 4. If enough inliers cannot be found, then the match is removed from the top BOVW results.

III. Methodology

Drawing from Spatial Pyramid's ability to retain some basic feature location information, swBOVW subdivides both the query and database images and seeks to find the best translation and scale match, to account for possible misalignments and differences in distances.

A. Module Insertion Location

RANSAC has proved successful in eliminating false positives, but as we have found in our implementation, at the cost of a higher number of `unsure' results. As presented in this paper, swBOVW can be used in three different ways to improve results. The first is as a replacement for RANSAC as the geometric validation method. The second is to run swBOVW in addition to RANSAC, only when RANSAC rejects enough of the top matches to cause an `unsure' result to occur. In that situation, a top match would only be rejected if it failed both the RANSAC and swBOVW algorithms. This strategy will avoid the computational cost of running swBOVW when not necessary. The third strategy is to reduce the number of database images down to two, one facing each end of the field, and then replacing the standard BOVW matching algorithm with swBOVW, supported by RANSAC.

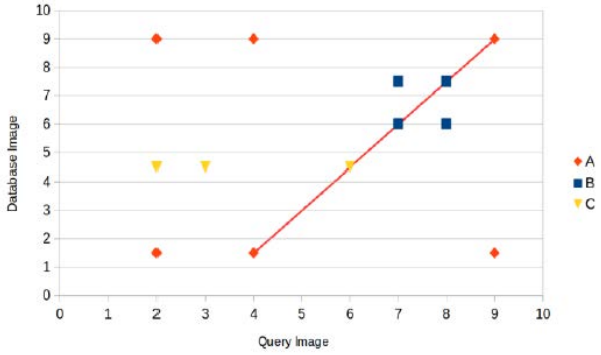

Figure 4. Scatter plot of features present in both the query and database images. Each occurrence of a feature in one image is plotted against all occurrences of that feature in the other image. Note that feature D does not occur in the database image (see Fig. 3), and so cannot be plotted on this graph. RANSAC will attempt to fit a line like the one shown in red, with each of the features occurring on the red line representing a matched feature between the two images.

B. Image Subdivision

After the features have been extracted by 1D SURF, the image strip is subdivided into sixteen equal divisions and a feature frequency vector is then constructed for each subdivision. Larger subdivisions can then be easily constructed by combining the frequency vectors of these foundational sixteen subdivisions. For labeling purposes, these sixteen divisions are called level one. Specific combinations of the level one subdivisions will generate other levels, with decreasing amounts of specific spatial information. The equations to generate levels 2, 3, and 4 are shown next:

$$L2_i = \sum_{j=i}^{i+1} L1_j, i \subseteq \{1, 2, ...15\} \quad (2)$$

$$L3_i = \sum_{j=i}^{i+2} L1_j, i \subseteq \{1, 2, ...14\} \quad (3)$$

$$L4_i = \sum_{j=i}^{i+3} L1_j, i \subseteq \{1, 2, ...13\} \quad (4)$$

These three different levels behave as a scale modifier, essentially enlarging one image in comparison to the other. To maximise runtime efficiency, the real-time query images

only calculate level three, while the database images precalculate all of the levels when they are first stored. Levels 2 through 4 are used rather than levels 1 through 3, as we found the finer resolution resulted in unnecessarily longer computation times.

C. Image Comparison

In our implementation, the matching of query and database images without applying scaling is done by comparing the level threes of both images. One of the images is given an initial offset of two L3 blocks. As can be seen in Fig. 5 and 6, this ensures that matching is being performed against at least half an image. After some preliminary testing, this choice provided the best trade-off between matching accuracy and computational effort. The offset image is then shifted along by one L1 subdivision at a time until it reaches the same offset on the other side of the image (see Fig. 5). At each stage along this translation, the corresponding blocks from each image have their feature frequency cosine similarity calculated, and the average is calculated in order to emphasise whole image matches rather than local maxima. To account for different levels of scaling, the L3 of the query image is then compared to the L2 and L4 of the database image (a scale factor of ⅔ and 1⅓, respectively). See Figure 6 for an illustration of the ⅔ case.

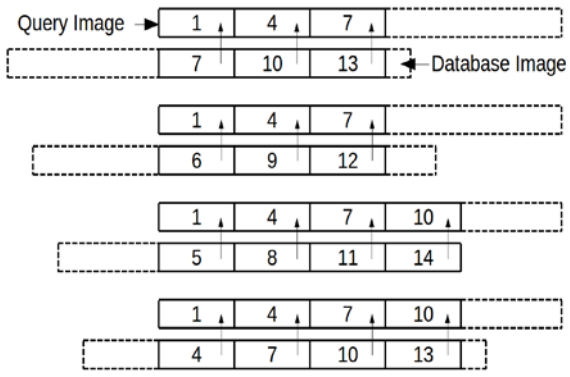

Figure 5. Illustration of image strip comparison with no scale modifiers using swBOVW. Since no scale modifiers are being used, the two image strips have been subdivided in the same way as defined in Section 3.2. As the database image is shifted (the first four steps are shown in this image), the corresponding subdivisions in the two image strips are compared to find their cosine similarity. The numbers represent which subdivisions in the image strips are being used, and the arrows indicate which subdivisions are being directly compared. For example, in the first (top) comparison, $L3_7$, $L3_{10}$, $L3_{13}$ from the database image are being compared to $L3_1$, $L3_4$, $L3_7$ from the query image, respectively. As the database image is shifted across, additional blocks will come become available to compare against the query, as can be seen in the third translation. The cosine scores are then averaged, after the full translation is finished, and the highest cosine average score is returned.

A minimum cutoff value of 0.3 for the cosine similarity was selected for the first two implementations of swBOVW, and a value of 0.2 for the third one, both based on empirical testing conducted when the method was being fine-tuned.

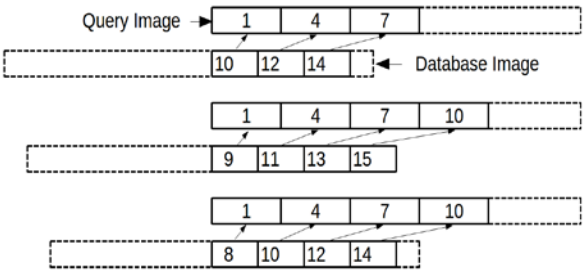

Figure 6. Illustration of image strip comparison with a ⅔ scale factor. To achieve this, the database image has been subdivided into smaller subdivisions, following the formula set out in Section 3.2. See Fig. 5's caption for more information.

D. Data Collection

The images used for queries and the database were captured in an indoor lab environment set up as a RoboCup robotic soccer field. For reference, the field is 6 meters long and 4 meters wide. For the first two swBOVW implementations, a total of eighteen database images were used, nine facing each end of the field and distributed as shown in Fig. 7. In the third implementation, only one image from the center of the mid-field line was taken for each field end, i.e. two database images. In addition, 32 query images were taken in each direction, for a total of 64, spaced evenly along the orange line shown in Fig. 7. The images were captured using a DARwIn-OP humanoid robot [14], while all testing was done on an Intel Core i7-6820HQ 2.70GHz. The wall behind the away goal was covered with many distinct features, while the wall behind the home goal was more homogeneous, as shown in Fig. 8.

E. Code Implementation

The base code used for 1D SURF and BOVW was taken from RoboCup's rUNSWift robotic humanoid soccer team's 2015 c++ code release[1]. rUNSWift placed first in their league at RoboCup's World Cup two years in a row in 2014 and 2015. Their code was designed for a different humanoid robot and used a different software architecture, so the code was reimplemented into NUbot's software framework [15]. rUNSWift had a few minor improvements implemented onto their humanoid robot for better real-time performance, such as combining the features found in two consecutive image frames, which were not applicable to this research. The images captured for this research were singular still-frames taken from various locations around the field, and could not benefit from consecutive frame improvements.

[1] github.com/UNSWComputing/rUNSWift-2015-release

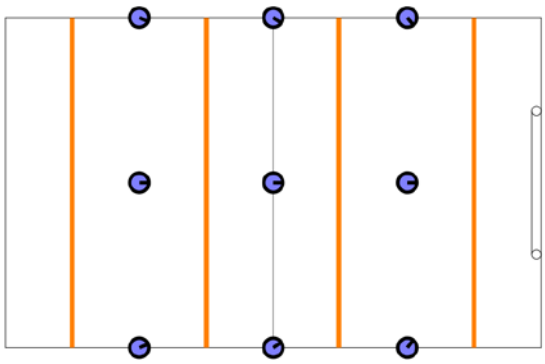

Figure 7. Diagram of soccer field where images were taken. The blue circles show where the 9 background images were taken for the away goal. The orange lines show where 32 evenly spaced query images were captured from. For the home goal side, everything is mirrored about the mid-field line.

(a) Away Goal (b) Home Goal

Figure 8. Examples of dataset images showing the two ends of the field. Note that the two ends contain several elements (chairs, tables, shirts, etc), thus simulating a real soccer playing field, where people and objects will be visible behind the goal posts

The feature vocabulary list used by BOVW clusters similar features into 170 distinct categories, and is the same that rUNSWift included in their 2015 code release. The cutoff value for minimum cosine similarity was kept the same as in the 2015 code release. However, the RANSAC minimum inlier cutoff (the minimum number of features RANSAC must be able to match) was optimised for this dataset to provide a competitive benchmark for swBOVW to be compared against.

IV. Results

Table I shows the results for six different 1D SURF + BOVW setups. The first two implementations of swBOVW are presented in the 18 image database section, while the third is shown in the 2 image database section. The performance of all three implementations of swBOVW are evaluated according to two criteria: accuracy and CPU run times. In the top section, the first result shows Bag-of-Visual-Words (BOVW) with RANSAC for the geometric validation (the benchmark), the second replaces RANSAC with swBOVW as the geometric validation method (1st swBOVW implementation), and the third combines RANSAC and swBOVW (but only when RANSAC causes an unsure result to be returned) (2nd swBOVW implementation). In the bottom section, the first result shows how the performance of BOVW + RANSAC degrades with a 2 image database, the second shows swBOVW replacing BOVW (1D SURF + swBOVW), and the third result extends the second by adding RANSAC (1D SURF + swBOVW + RANSAC) (3rd implementation). Accuracy has been broken down into true positives, false positives, and unsure. When presented in the table below, these three measures are shown in the same order, separated by forward slashes. The total of these three measures will always add up to 32, the number of query images for each field end. The results are split into query images for the away goal and for the home goal, along with a combined total. The last column shows the average CPU run time per query image.

To provide the benchmark by which the three swBOVW implementations will be measured against, the first row of results in the table show RANSAC used as the geometric validation method. This method is very effective at ensuring no false positives, but comes at the cost of an 'unsure' image rate of 35% (23 unsure images).

The first use of swBOVW replaces RANSAC as the geometric validator. This has resulted in a 30% reduction in unsure results, down to 16 (occurrence rate of 24%). However, this has come at the cost of 1 false positive image (rate of 1.5%). The computational cost of this improvement is 32ms on average (an 80% increase).

In the second case, adding swBOVW to RANSAC in the situation where RANSAC creates an unsure result (22% of cases), has resulted in a considerable 48% reduction in unsure results down to 12 images (occurrence rate of 18%). The computational cost of this improvement is 11ms on average (a 27% increase), which is considerable better than the first swBOVW implementation. However, the 1 false positive image remains (rate of 1.5%).

For the third implementation to be viable, the number of database images is reduced down to two. The first result in the 2 image database section of the table shows how the 18 image benchmark degrades. The `unsure' results increases by 25% to 30 images (occurrence rate of 45%). The third implementation replaces the standard BOVW with swBOVW. However, on its own, this change does not constitute an improvement to the 18 image BOVW + RANSAC benchmark, with 2 false positive images, and 28 unsure images. However, with RANSAC providing a second geometric validation, swBOVW is able to achieve comparable results to the 18 image benchmark, with 24 `unsure' results, but can do it for half the computational cost.

V. Conclusions

In this paper we presented a geometric BOVW extension called *sliding window BOVW* (swBOVW), with application to computationally constrained robotic image matching applications. Three alternative implementations of swBOVW were shown, either replacing the commonly used geometrical validation method of RANSAC or complementing it, improving performance each time.

In our first implementation of swBOVW, by replacing RANSAC with swBOVW, we found we were able to lower the unsure results by 30%. This improvement did however

come with an 80% increase in computational cost, which is enough to require a case-by-case decision on whether the intended robotic platform is able to afford the increased cost required to achieve the increased performance.

In the second case, by adding swBOVW to the RANSAC geometric validation method only in cases were RANSAC failed, we were able to further improve the unsure image reduction to 50%. In addition, we were able to dramatically lower the computational cost of swBOVW, achieving a relatively small 27% computational cost increase and is one of the key breakthroughs in our results. This has allowed us to take full advantage of what otherwise would have been a prohibitively slow method. Being used together like this has enabled the strengths of each geometric validation method to be emphasised, while simultaneously compensating for their weaknesses.

TABLE I. RESULTS

18 image database

Method	Query Image (true pos./false pos./unsure) Away goal	Home goal	Total	CPU time
BOVW + RANSAC [10]	21/0/11	20/0/12	41/0/23	40ms
BOVW + swBOVW	23/0/9	24/1/7	47/1/16	72ms
BOVW + RANSAC + swBOVW	25/0/7	26/1/5	51/1/12	51ms

2 image database

Method	Query Image (true pos./false pos./unsure) Away goal	Home goal	Total	CPU time
BOVW + RANSAC [10]	17/0/15	17/0/15	34/0/30	7ms
swBOVW	15/2/15	19/0/13	34/2/28	15ms
swBOVW + RANSAC	21/0/11	19/0/13	40/0/24	20ms

The false positive rate of 1.5% obtained in the first two implementations of this method is arguably minor, and can be mitigated through a variety of means when implemented. In the case of a soccer robot, where an image stream is being analysed in real-time, localisation errors caused by spurious misclassifications would be corrected almost immediately due to the very high rate of correct classifications (80%).

If a particular application of swBOVW had an acceptable accuracy level for the 18 image benchmark, but required a reduction in computational cost, then our third implementation is able to achieve a comparable accuracy but for half the computational cost. This implementation reduces the number of images in the database down to two, and replaces the traditional BOVW with swBOVW, while using RANSAC to provide a second geometric validation.

Future extensions of this work will involve investigating the effectiveness of our one dimensional geometric validation technique on standard two dimensional images in situations where there is little image rotation. Additionally, while this work focused on disambiguating field ends, it could also be used in more general image-based localisation methods.

While most research focuses on taking advantage of the increasing computational power being made available to software engineers, there will always be a place for extremely lightweight algorithms to run unobtrusively in the background providing valuable secondary data. This is particularly true in the area of small, unmanned aerial vehicles and terrestrial robots - and becomes a critical issue in the emerging area of microrobotics.

REFERENCES

[1] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in Proceedings Ninth IEEE International Conference on Computer Vision, Oct 2003, pp. 1470–1477 vol.2.

[2] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval, ser. MIR '07. New York, NY, USA: ACM, 2007, pp. 197–206. [Online]. Available: http://doi.acm.org/10.1145/1290082.1290111

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov 2004. [Online]. Available:https://doi.org/10.1023/B:VISI.0000029664.99615.94

[4] H. Bay, T. Tuytelaars, and L. Van Gool, SURF: Speeded Up Robust Features. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. [Online]. Available: https://doi.org/10.1007/11744023_32

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, 2006, pp. 2169–2178.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysisand automated cartography," Commun. ACM, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[7] V. Ptrucean and M. Ovsjanikov, "Affine invariant visual phrases for object instance recognition," in 2015 14th IAPR International Conference on Machine Vision Applications (MVA), May 2015, pp. 14–17.

[8] J. Tang, D. Zhang, Y. Zhang, and Q. Tian, "Region similarity arrangement for image retrieval," in 2016 IEEE International Conference on Multimedia and Expo (ICME), July 2016, pp. 1–6.

[9] H. Zhao, Z. Nong, P. Liu, and Q. Li, "Saliency weight and local quadrant constraint for visual search," in 2015 8th International Congress on Image and Signal Processing (CISP), Oct 2015, pp. 561–566.

[10] P. Anderson, Y. Yusmanthia, B. Hengst, and A. Sowmya, Robot Localisation Using Natural Landmarks. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 118–129. [Online]. Available: https://doi.org/10.1007/978-3-642-39250-4_12

[11] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in Asian Conference on Computer Vision. Springer, 2010, pp. 709–720.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on IEEE, 2007, pp. 1–8.

[13] P. Anderson, "New methods for improving perception in robocup SPL," Undergraduate Honours Thesis, The University of New South Wales, 2012, unpublished.

[14] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong, "Development of open humanoid platform DARwIn-OP," in SICE Annual Conference 2011, Sept 2011, pp. 2178–2181.

[15] T. Houliston, et al., "Nuclear: A loosely coupled software architecture for humanoid robot systems," Frontiers in Robotics and AI, vol. 3, p. 20, 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/frobt.2016.00020