Using TextSpotting to Query the World.

Ingmar Posner and Peter Corke and Paul Newman

***Abstract*— The world we live in is well labeled for the benefit of humans but to date robots have made little use of this resource. In this paper we describe a system that allows robots to read and interpret visible text and use it to understand the content of the scene. We use a generative probabilistic model that explains spotted text in terms of arbitrary search terms. This allows the robot to understand the underlying function of the scene it is looking at, such as whether it is a bank or a restaurant.**

We describe the text spotting engine at the heart of our system that is able to detect and parse wild text in images, and the generative model, and present results from images obtained with a robot in a busy city setting.

I. INTRODUCTION

Text, by design, is a valuable resource that carries very strong semantic information that cannot otherwise be inferred from other sensing modalities. In this paper we describe a system to exploit this valuable and under-exploited source of information for robots.

Our earliest experiments indicated that text is indeed plentiful — street signs, bus stops, and shop fronts all provide good quality text that is rich in information about function and about location. Shop fronts are particularly rich in text that provides information about the function of the shop and is also potentially queryable using internet search resources to determine its location. Street signs provide important navigational cues. Key words like "push" or "pull" are indicative of doors, and so on. There is also a surprising amount of mobile text in the world as we learnt from the experiment just described. That is text that is moves with respect to the environment and includes car registration plates, the fronts of buses (which have place names), advertising on the sides of buses and vans, and logos on shopping bags and clothing. Therefore it cannot be assumed that text is necessarily a label associated with the place where the text was seen.

In this paper we describe an implementation of this idea. The core of our system is the textspotting engine which robustly detects and reads text in the environment, see Figure 1. Despite the long history of automatic text recognition we discovered that the application beyond printed documents is a current research problem. The challenges with wild text include the lack of contrast between text and its background, the rich diversity of fonts and character sizes, highly variable horizontal and vertical alignment of characters and related words, and geometric distortion due to non fronto-parallel viewing.

I. Posner and P. Newman are with the Mobile Robotics Group, Department Of Engineering Science, Oxford University, UK ingmar@robots.ox.ac.uk, paul@robots.ox.ac.uk

P. Corke is with the School of Engineering Systems, Queensland University of Technology, Australia peter.corke@qut.edu.au

Fig. 1. A typical example output of our text spotting pipeline.

Fig. 2. The data acquisition robot used in this work. Images were captured using the Bumblebee camera mounted on a pan-tilt head.

In this work we are concerned with understanding the underlying topic behind one or more observed words. For example a restaurant might (if we were really lucky) be indicated by the observed word "restaurant", but it may also be indicated by synonyms such as "bistro" or words that denote the cuisine ("Chinese", "Thai") or the food specialty ("seafood", "pizza", "steak"). We describe a generative probabilistic model that explains spotted text in terms of arbitrary search terms.

The contributions of this work are a robotic system which exploits a valuable but unused navigational and informational resource using vision and OCR, and a generative model that explains the subject of the scene in terms of detected text. The remainder of this section describes related prior

work. The core components of the textspotting engine, text detection and optical character recognition (OCR) are described in Section II. The generative probabilistic model that we use to select images relevant to arbitrary search terms is described in Section III. Our conclusions and current research directions are summarized in Section V.

A. Related Work

The use of OCR with robots is suggested, but not implemented, in [1]–[3]. A book manipulation robot [4] uses OCR to confirm the title of the book to be taken from a shelf, and [5] describe an indoor mobile robot that performs OCR although the extracted text is not utilized. In 1994 Engel *etal* proposed a small robot with onboard DSP-based computation that would read signs and licence plates but it is not clear how far this work progressed [1]. A 2003 paper by Mirmehdi *etal* about OCR proposed its application to robotic navigation [2].

An important part of our system is the application of OCR to outdoor scenes and this is an area of current research interest. ICDAR[1] has organized two competitions (2003 and 2005) for the robust detection of wild text based on a standard set of of labelled images. The results are summarized in [6], [7]. Other non-document OCR applications include detecting text in television streams [8], licence plate recognition [9]–[11], and assistive devices for the visually impaired [12], [13].

II. TEXT SPOTTING TOOL CHAIN

The heart of our system is the text spotting engine. Commonly this problem is decomposed into stages: the detection of text in the image, recognition of characters, and then grouping of characters into coherent units of text (such as words or sentences). With few exceptions (see, for example, [14]) these individual steps are considered independent sequential processes and no information is shared between between them.

Our textspotting implementation follows this classical approach to the problem and the principal elements are:

1) Text detection. Determine regions of the input image that are likely to contain text.
2) Optical character recognition (OCR). Convert these image regions to character strings, typically words.
3) Layout analysis. Concatenate the strings from spatially adjacent regions into sentences. Boxes with similar sized characters that are close and aligned, horizontally or vertically, are merged.
4) Text filtering and spelling correction. The output from the OCR stage is very noisy, often containing spurious characters and many character substitution errors.

A. Detecting Text in Natural Scene Images

The aim of this stage is to efficiently detect instances of text in a given image. Boosting techniques [15] coupled with an attentional cascade, introduced in [16], provide a straightforward means to this end and have a successful track record in text detection [8], [17], [18]. In this work we apply GentleBoost [19] with the base classifiers consisting of decision stumps operating on a set of Haar-like features. These features are obtained by sliding predefined block patterns over an image and computing features as functions of statistics of such as mean and variance of each of the individual blocks.

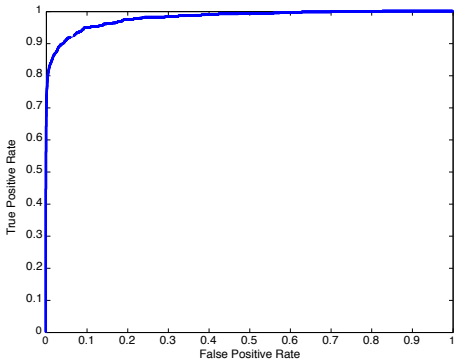


Fig. 3. Performance of a single boosted classifier after 1000 rounds of training using both the training partition of the ICDAR data and the Weinman data.

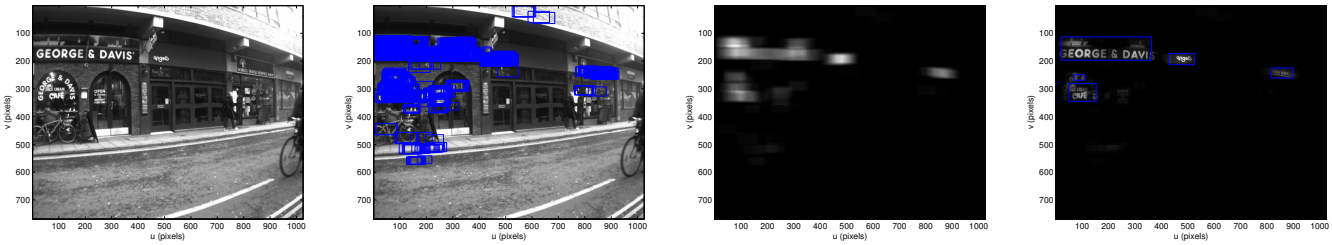
Chen et al [17] note that image gradient information captures a distinctive characteristic of text. We follow [18] in our selection of features and use feature channels based on x- and y-gradient, gradient magnitude in addition to mean and variance. In summary, we compute 22 features from each of five feature channels giving a total of 110 feature dimensions to be considered.

We employ two independent third-party data sets for training of our classifier cascade. The first dataset is provided publicly as part of the ICDAR 2003 challenge on robust reading and text location[2]. It consists of a training and a test set each comprising 250 hand-labelled images drawn from indoor and outdoor environments. Since our focus is on outdoor applications we augmented this data with a subset of the data used by Weinman [14] comprising 300 images taken in outdoor urban settings and include a higher proportion of natural scene clutter as well as instances of multiple lines of text per label.

To investigate the efficacy of the features, we trained and evaluated a single monolithic boosted classifier. Training was based on 450 positive and 2000 negative examples of text randomly sampled from a combination of the *training partition* of the ICDAR data and the complete Weinman data. The trained classifier was evaluated using a hold-out set of 996 positive and 38,000 negative data sampled from the same datasets. The classifier performance on the validation set after 1,000 rounds of training is presented in Figure 3. The number of training rounds was set arbitrarily large, designed to guarantee convergence to a stable validation error. This performance plateau, however, was typically observed to be

[1]International Conference on Document Analysis and Recognition

[2]http://algoval.essex.ac.uk/icdar/RobustReading.html



(a) (b) (c) (d)

Fig. 4. Stages of the text spotting pipeline. (a) the original image, (b) with overlaid detection rectangles for scales 48, 57 and 69 (c) the text likelihood map, (d) the detected text regions for this scale range.

reached after only a few hundred rounds of training. Figure 3 indicates an adequate separation of the classes.

In order to provide an efficient classification framework with a suitably low false positive rate we deploy a cascade of boosted classifiers rather than a single monolithic one. The training was conducted using text regions randomly sampled from a combination of the *training partition* of the ICDAR data and the complete Weinman data. Each stage of the cascade was trained using 400 positive and 1000 negative examples. The negatives are continuously sampled out of a pool of 35,000 data. The validation set consisted of 1046 positive and 5000 negative examples. The first five levels only leverage a single digit amount of features, while levels six and seven use 18 and 108 features, respectively. Also note that the final output of the cascade yields a relatively high detection rate (79.4%) while only 1.6 out of a thousand detections are spurious.

B. Region extraction

The output of the previous stage are lists of rectangles, one list for each scale, which are classified as containing text, see Figure 4(b). A typical image will have hundreds of rectangles at each of a number of scales. The rectangles are overlapping and at each scale we look for rectangles that have support, that is they overlap with at least N other rectangles (we use $N = 3$). It is highly unlikely that wild text will match the scale steps exactly so we consider the supported rectangles in a sliding window of M adjacent scales (we use $M = 3$). Each rectangle votes for the pixels that it contains and the votes are tallied in a voting array the same size as the original image, see Figure 4(c). The voting array is thresholded at 25% of the maximum value and bounding boxes for the regions computed. The selected regions, at this scale, are shown in Figure 4(d).

Good bounding boxes are important for success in subsequent stages of the pipeline, and our current approach too often gives bounding boxes that are too small or too large.

C. Optical character recognition

Today OCR packages are low-cost and very reliable for printed text which exhibits high contrast, simple background, uniformity in font and character size, and horizontal alignment of characters — characteristics not shared by wild text. Most commerical OCR packages are intended for integration with desktop word processing tools not robots and we evaluated two open-source OCR packages: GOCR and Tesseract [20] and chose the latter. Tessearct deals well with skewed baselines which is advantageous when dealing with geometric distortion due to non fronto-parallel viewing and avoids the need to rectify image regions.

The main mode of failure is misrecognition of characters and intercharacter spacing. Single character substitution errors are common (eg. zero for oh, one for ell, five for ess). Spaces can appear between adjacent characters, or spaces between words are sometimes not seen — both cases are problematic. The root cause is the wide range of fonts that are found in outdoor signage.

D. Probabilistic Error Correction

The output of the OCR engine can be improved considerably when the output is constrained to some set of meaningful words. Simple dictionary checks would discard any word not found but this is unsatisfactory for the case of common single character substitution errors. Instead we use probabilistic inference over the true word present in the scene, w, given a possibly erroneous detection, z , $p(w|s)$.

Let $\mathcal{Z}$ denote the set of all possible OCR detections such that $z \in \mathcal{Z}$. Furthermore, let $\mathcal{V}$ denote the set of all terms in the English language such that $w \in \mathcal{V}$. We think of z as a noisy translation of some unknown generating word w. The posterior distribution over all words in the set $\mathcal{V}$ can be expressed as

$$p(w|z) = \frac{p(z|w)p(w)}{p(z)} \quad (1)$$

$$= \frac{p(z|w)p(w)}{\sum_{w \in \mathcal{V}} p(z|w)p(w)} \quad (2)$$

Evaluation of this expression requires the determination of $p(z|w)$ — the distribution of text detections given a correctly spelt and complete observation-generating word w. Intuitively, the "closer" z is to a word, the more likely that word is to explain that detection. We use the Levenshtein edit distance $\phi(z, w)$ to capture this sense of distance between detected text z and word w and write

$$p(z|w) = \alpha \quad e^{-\alpha\phi(z,w)}. \quad (3)$$

Here α is a free parameter encoding the accuracy of the text detection system. For the results presented in this paper α was set by hand using random spelling mistakes. No data

Fig. 5. Examples of wild text found by the robot. The annotations are the raw Tesseract output without any error correction applied.

Fig. 6. Examples of wild text found by the robot after error correction.

Fig. 7. Examples of incorrect reading or processing of wild text found by the robot.

contained either in the training or test sets were used. In future work we intend to learn this parameter from a large training set. Finally, Equation 2 requires the specification of the prior probability of a given word w occurring in a scene. We use word frequencies obtained from the British National Corpus [21], a collection of approximately 100×10^6 words encompassing ca. $130,000$ unique terms.

In the particular streetscape that comprises this dataset there are a number of shops with French names or products which will be incorrectly corrected.

III. Relating Text to Subjects

We derive a model which explains the subject of an image in terms of the detected text it contains. Importantly, because of the use of a large corpus of text, we need not limit ourselves to a finite set of subjects chosen *a-priori*. We apply this model to execute subject searches in which a robot will return a list of places and views which relate semantically to the search term. Specifically, we require that searching for the subject *mobile phone* would return coordinates of views containing text like "nokia", "samsung", "broadband", "3G" etc. — evidence that the scene captured in an image has something to do with mobile phones. Note that we do not expect or demand flawless text detection since, due to the detector model introduced in Section II-D, we can handle incorrect detections like "nqkio", "smssag", "roodbond" and "3O".

Given a corpus of images, let $\mathcal{Z}$ denote the set of all detections of text throughout the corpus. Furthermore, let $\mathcal{S}$ denote the set of all possible scene subjects. Our goal is to explain a particular subject term $s \in \mathcal{S}$ with respect to a given particular text detection $z \in \mathcal{Z}$. In a probabilistic sense we can express this as the task of finding the posterior probability of the search term given the detection

$$p(s|z) = \frac{p(z|s)p(s)}{p(z)}. \tag{4}$$

The partition function $p(z)$ is the probability distribution over all possible detections and can be expanded in terms of a marginalization over subject terms of the joint distribution $p(z, s)$. If we take all subjects to be equally likely, Equation 4 reduces to

$$p(s|z) = \frac{p(z|s)p(s)}{\sum_{s\in\mathcal{S}} p(z|s)p(s)}. \tag{5}$$

$$= \frac{p(z|s)}{\sum_{s\in\mathcal{S}} p(z|s)}. \tag{6}$$

The term $p(z|s)$ is the likelihood of the OCR returning a string z when the underlying scene subject is s. We leverage the detector model introduced in Equation 3 to account for

Fig. 8. Images related to the topic "lunch".

the noise in the detection and parsing of text. We introduce a layer of now hidden variables $w \in \mathcal{V}$, where once again $\mathcal{V}$ denotes the vocabulary of the English language and each w is a word. By marginalising over the $\mathcal{V}$ our desired likelihood term $p(z|s)$ can be expanded in terms of the hidden words

$$p(z|s) = \sum_{w \in \mathcal{V}} p(z|w, s)p(w|s). \quad (7)$$

If we take detection noise to be independent of subject, we can express the likelihood $p(z|s)$ as

$$p(z|s) = \sum_{w \in \mathcal{V}} p(z|w)p(w|s). \quad (8)$$

which requires the determination of the detector model $p(z|w)$. The remaining term in Equation 8 is $p(w|s)$ — the probability of a bonafide word w occurring in a corpus of words on subject s. We assume an internet connected robot and launch a web search for the subject string s and aggregate the words in the returned documents into a single *subject document*. For the results in this paper we searched the websites of the BBC News, The New York Times and the Guardian Newspaper. The construction of the subject document allows $p(w|s)$ to be estimated directly by counting the number of times word w occurs.

IV. EXPERIMENTAL RESULTS

We used the robot Marge, an iRobot ATRV-JR equipped with a variety of sensors, see Figure 2. Images are captured with a Bumblebee stereo head that provides 1024×768 greyscale images with a $60\,\mathrm{deg}$ field of view. In this analysis we consider only images from the left camera in the stereo pair. Future analysis will look at using 3D structure to improve text segmentation by looking for a local plane in the region indicated by the text detectio stage. For this we would use either stereo disparity from the camera or the 3D point cloud collected by the nodding laser which can be seen on the middle deck of the robot in Figure 2.

Figures 5 and 6 shows a small selection of typical results of applying our text spotting pipeline to the collected dataset of 941 images. Figure 5 shows a number of images with successfully detected words. Figures 5 presents the raw OCR output before error correction is applied. Note that a number of words are misspelt and that, for the middle two frames,

Fig. 9. Images related to the topic "taxi".

Fig. 10. Images related to the topic "bank".

the bounding box has truncated a word. Figure 6 shows how our system recovers some of the mispelt words or discards those that were truncated. As well as the extracted words the system provides a confidence level $p(w|z)$ — computed as per Equation 2 — as to how likely the inferred word w explains the observation z . This posterior probability over generating words provides a natural and intuitive way of thresholding system output. Figure 6 only shows detections with a confidence greater than 90%.

The failure cases shown in Figure 7 are interesting and shows examples of what we call texture words. In this case the texture has come from vertical window edges and bricks, but other architectural features and adornments also generate texture words. The texture has elicited a positive response

lunch		*taxi*		*bank*	
term	$p(s\|z)$	term	$p(s\|z)$	term	$p(s\|z)$
restaurant	**0.0186**	**telephone**	**0.0112**	**barclays**	**0.1131**
barclays	0.0052	queue	0.0092	george	0.0060
queue	0.0035	february	0.0051	street	0.0047
children	0.0033	street	0.0042	february	0.0043
keep	0.0032	over	0.0024	telephone	0.0041

TABLE I

THE TOP 5 WORDS AUTOMATICALLY EXTRACTED FROM THE DATASET RANKED IN TERMS OF LIKELIHOOD. THOSE SHOWN IN BOLD FONT ARE ABOVE THE THRESHOLD.

from the text detection stage and the OCR stage has then done its best to find characters. Typically texture results in the letters from the set "ILETUCMWA".

We applied our topic finding model to this set of images, querying in turn for the subjects *lunch, taxi* and *bank*. In the first instance the output of the system consists of a ranking of all the terms extracted from the corpus of images based on the posterior probability $p(s|z)$. The top five returns per subject are shown in Table I together with the probability of the topic given the observed word. In every case the system manages to successfully extrapolate from the query to semantically related terms, and we apply a threshold at 1%. The images corresponding to our query terms are shown in Figures 8 – 10.

V. CONCLUSIONS

We have described a robotic system that is capable of detecting and reading wild text, the semantically rich textual cues placed in man-made environments. This is a rich resource for robotics and we have demonstrated its potential for topic-based navigation. We have shown how a human-meaningful topic can be used to identify a relevant image, and conversely how an image can be mapped into a set of topics.

This is early work in the field of literate robotics and our work is progressing on several fronts. Firstly we are integrating the systems into a 3G-connected robot that can implement these techniques online. Secondly we are investigating means to improve the performance of the OCR step which is currently exhibiting a very high error rate, and apply learning or adapation to the text voting stage. Thirdly we are investigating how wild text can be used in conjunction with an internet-based geocoding service to provide a spatial likelihood function that can be used directly or fused with other localization modalities.

VI. ACKNOWLEDGMENTS

The authors would like to thank Jerod Weinman for making his data available for use in this work. The work reported here was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.

REFERENCES

[1] G. Engel, D. Greve, J. Lubin, and E. Schwartz, "Space-variant active vision and visually guided robotics: Design and construction of a high-performance miniature vehicle," in *INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION*, pp. 487–487, IEEE COMPUTER SOCIETY PRESS, 1994.

[2] M. Mirmehdi, P. Clark, and J. Lam, "A non-contact method of capturing low-resolution text for OCR," *Pattern Analysis & Applications*, vol. 6, no. 1, pp. 12–21, 2003.

[3] A. Carbone, A. Finzi, A. Orlandini, F. Pirri, and G. Ugazio, "Augmenting situation awareness via model-based control in rescue robots," in *Proc. of IROS-2005 Conference*, Citeseer, 2005.

[4] R. Ramos-Garijo, M. Prats, P. Sanz, and A. Del Pobil, "An autonomous assistant robot for book manipulation in a library," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3912–3917, 2003.

[5] J. Samarabandu and X. Liu, "An edge-based text region extraction algorithm for indoor mobile robot navigation," *International Journal of Signal Processing*, vol. 3, no. 4, pp. 273–280, 2006.

[6] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, vol. 2, pp. 682–687, Citeseer, 2003.

[7] S. Lucas, "ICDAR 2005 text locating competition results," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition, ICDAR05*, pp. 80–84, Citeseer, 2005.

[8] M.Lalonde and L. Gagnon, "Key-text spotting in documentary videos using adaboost," in *Proceedings of SPIE*, vol. 6064, pp. 507–514, 2006.

[9] Y. H. T. Wing Teng Ho, Hao Wooi Lim, "Two-stage licence plate detection using gentle adaboost," in *First Asian Conf. on Intelligent Information and Database Systems*, 2009.

[10] N. Ben-Haim, "Task specific image text recognition," Master's thesis, University of California, San Diego, 2008.

[11] L. Dlagnekov, "Video-based car surveillance: License plate, make, and model recognition," Master's thesis, University of California, San Diego, 2005.

[12] X. Chen and A. Yuille, "A time efficient cascade for realtime object detection: with applications for the visually impaired," in *Proceedings of the CVAVI05, IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2005.

[13] S. Hanif, L. Prevost, and P. Negri, "A cascade detector for text detection in natural scene images," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, 2008.

[14] J. J. Weinman, *Unified Detection and Recognition for Reading Text in Scene Images*. PhD thesis, University of Massachusetts Amherst, 2008.

[15] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting," *Jrnl. of Computer and System Sciences*, vol. 1, no. 55, pp. 119–139, 1997.

[16] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *Intl. Jrnl. of Computer Vision*, vol. 2, no. 57, pp. 137–154, 2004.

[17] X. Chen and A. L. Yuille, "Detecting and Reading Text in Natural Scenes," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 366–373, 2004.

[18] S. Escalera, X. Baró, J. Vitrià, and P. Radeva, "Text Detection in Urban Scenes," in *Proc. Conf. on Artificial Intelligence Research and Development*, pp. 35–44, 2009.

[19] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *Annals of Statistics*, vol. 28, 1998.

[20] R. Smith, "An overview of the Tesseract OCR engine," in *In Proc. of Intl. Conf. Document Analysis and Recognition (ICDAR)*, vol. 2, pp. 629–633, 2007.

[21] J. H. Clear, "The British national corpus," in *The digital word: text-based computing in the humanities*, pp. 163–187, Cambridge, MA, USA: MIT Press, 1993.