OCRAPOSE: AN INDOOR POSITIONING SYSTEM USING SMARTPHONE/TABLET CAMERAS AND OCR-AIDED STEREO FEATURE MATCHING

Hamed Sadeghi[⋆] *Shahrokh Valaee*[⋆] *Shahram Shirani*[†]

[⋆] University of Toronto, ECE Department, hamed.sadeghi@mail.utoronto.ca, valaee@comm.utoronto.ca
[†]McMaster University, ECE Department, shirani@mcmaster.ca

ABSTRACT

In this paper, we propose an image-based localization system, applicable for a number of indoor scenarios including office buildings, airports, chain stores, etc. In such applications, text/numbers are suitable distinctive landmarks for localization. The proposed system takes advantage of OCR to read the text/numbers and provide a rough estimate using the floor plan. Next, it performs OCR-aided stereo feature matching to refine the estimate by solving a PnP problem. Experiments show that this system achieves a median localization error of less than 50 cm for test positions located as far as 7 meters from a $20cm$ by $30cm$ number plate using different test devices in a university building scenario.

Index Terms— Indoor localization, OCR, PnP problem, Stereo feature matching, Coplanar features

1. INTRODUCTION

The most prevalent technology for indoor localization is based on fusion of Wi-Fi RSS fingerprints with inertial sensors data [1, 2]. In scenarios, where Wi-Fi access points are not available or whenever greater accuracy is required, image-based methods can be used as an alternative solution for localization. Image-based methods proposed in the literature can be categorized into two classes, image retrieval-based (fingerprinting-based) [3–6] and landmark-based (logo-based) [7,8] methods. Image retrieval-based methods are good candidates for localization in scenarios with no (or few) distinctive landmarks, such as outdoors or generic indoor scenarios. On the contrary, landmark-based methods provide good location accuracy wherever there exist enough number of highly textured and distinctive landmarks, such as shopping malls abundant with commercial logos [5,6]. Texture is required to extract enough feature points necessary for unique logo detection.

We have observed that in some indoor environments such as airports, chain stores or university (office) buildings, where the most distinctive landmarks are text and/or numbers, the aforementioned methods usually fail. The reason is that text or numbers are not as textured as commercial logos. Hence, stereo feature matching in landmark-based methods fails to extract enough distinctive features in order to distinguish between different numbers. For instance, feature-based methods cannot distinguish between room numbers 4148 and 4140 or gate numbers $B42$ and $B43$. There is also a high probability of seeing similar (repeated) visual scenery from different locations. Hence, common image retrieval-based methods might also fail to find the actual corresponding image (location) in these scenarios. This motivates the use of OCR engines to exactly recognize the existing numbers.

By recognizing the existing numbers in the query image, OCR can provide a rough user location estimate based on a floor plan.

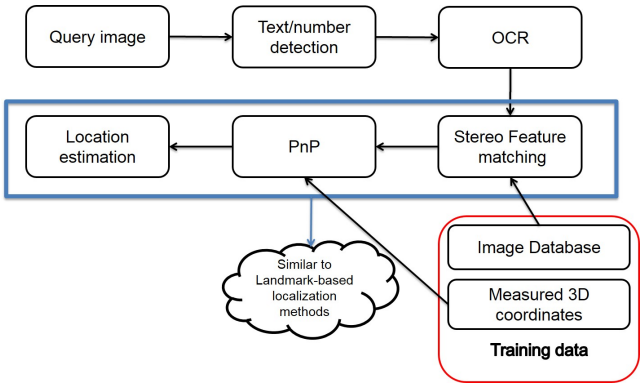


Fig. 1: Block diagram of the proposed localization system

To the best of our knowledge, [9] is the only work in the literature that utilizes OCR for rough localization in indoor scenarios. They combine OCR with magnetic tracking methods to provide better accuracies.

Our main contribution is to propose a system, called OCRAPOSE, based on a combination of OCR and landmark-based techniques (stereo feature matching + PnP) for *fine* localization in the mentioned applications. In fact, OCR provides a rough location estimate and the following OCR-aided landmark-based method refines it. The proposed system is not a simple concatenation of OCR and landmark-based techniques. In fact, OCR improves the feature matching in three aspects. First, it confines the database search space to the images taken from the existing text/number plate. Second, it maximizes the probability of feature points co-planarity, which is required for query features 3D labelling. Lastly, it decreases the probability of having cross matched outliers since the vicinity of the text/number plate in query image is matched to the vicinity of the same text in the database image. State of the art methods are used to design different parts of the system, including text/number detection, OCR, stereo matching, PnP, etc.

2. THE PROPOSED SYSTEM (OCRAPOSE)

Fig. 1 depicts the structure of the proposed system. In the sequel, we explain the role of each block in the system in details.

2.1. Text/number detection block

As seen in Fig. 1, first, the query image enters the text/number detection block. The role of the text/number detection block is to detect the regions of interest (RoIs) that contain the number. The goal is

NOT to miss the existing text/number plate regions while having the ability to reject false positives (non-number plate regions detected as number plates).

In this block, we first increase the query image contrast by adjusting its histogram as a preprocessing stage to prepare it for number regions detection. Among several experiments with cellphones cameras in the university buildings, we found that graph-based visual saliency (GBVS) maps proposed by [10] almost never misses the number plates in RoI detection. Fig. 2a shows the GBVS map of a test image taken inside a university building. As seen, there are high peaks located inside the plate as desired and other false positive peaks located at the notice paper, door handle and corner of the garbage can. The next task is to locate the regions of interest.

In order to locate the RoIs, we find the peaks (local maxima) in the saliency map. This map amplitudes are normalized to 1. First, we vectorize the map signal to be a vector X with length N, where N is equal to the number of image pixels. Then, we find the locations (L_i) of the peaks in the signal. We impose the following conditions on number of peaks (N_{peaks}), peaks height (i.e. their value in the GBVS map, H_i) and their locations

- $N_{peaks} \leq 10$
- $H_i \geq 0.1 \qquad \forall i$
- $|L_i - L_j| > \frac{N}{20} \qquad \forall i, j$

where parameters have been found empirically. The last condition states that the distance of any two peak locations (i.e. L_i and L_j) should be at least $\frac{1}{20}$ of the vectorized map length. This is to prevent the case of having peaks close to each other, which belong the same salient object with a high probability.

Once peaks are selected and localized in the image ($L_i \rightarrow L_i^{Image} = (row_i, col_i)$), we search for rectangular regions around the detected peaks so they include rectangular text/number plates. These regions are selected in the following way. Consider a point X_k inside the map, with height H_k and image location $L_k^{Image} = (row_k, col_k)$. This point is considered to be inside the region of the peak P_i if

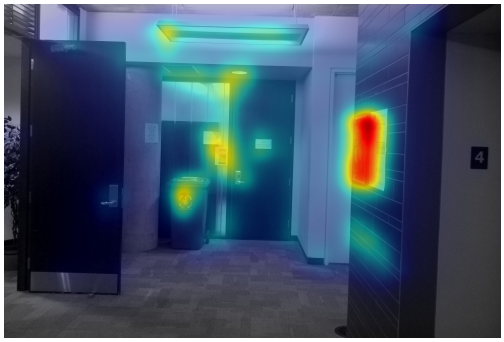
- $H_k \leq \frac{1}{2} H_i$
- $|row_k - row_i| \leq \frac{m}{scale_{row}}$
- $|col_k - col_i| \leq \frac{n}{scale_{col}}$

where m and n are the number of rows and columns in the image, respectively. $scale_{row}$ and $scale_{col}$ are two parameters that are found empirically. They basically show the ratio of the dimensions of the regions with respect to the image size.

The first condition usually prevents the RoIs to be perfectly rectangular since the saliency map surface could have any shape and its thresholding (first condition) results in a general form of RoI. Hence, RoIs are semi-rectangular as seen in gray in Fig 2b.

In our experiments, we realized the proposed RoIs have smaller area compared to circular ones (i.e. constant distance from the peak image coordinates $|L_k^{Image} - L_i^{Image}| < cte.$) or those resulted from pure height thresholding of the saliency map. Smaller regions mean better confinement to the number region and greater performance of the following recognition stage.

In our application, number plates are made of metal. Hence, ambient light experiences a uniform reflection from these surfaces and creates distinct patch-like regions with almost the same intensities in the query grayscale image. Such regions are good candidates to be detected as maximally stable extremal regions (MSER) [11] regions as also suggested by [12] for text region detection. Hence,



(a)

(b) (c)

Fig. 2: Results of Number detection and recognition, (a) Graph-based visual saliency map [10], (b) MSER regions inside the semi-rectangular RoIs close to the number plate, (c) The recognized number and the its box

we search for MSER regions in the detected RoIs. This would reject some of the false positive regions (RoIs without any MSER regions) and refine the number part of the ROIs, which improves the performance of the OCR block. Fig. 2b shows the MSER regions inside the detected semi-rectangular ROIs. Even after applying the MSER detector, some false positive regions will remain as seen on the garbage can in Fig. 2b. OCR block will reject these remaining regions afterwards since they do not contain any relevant number.

2.2. OCR block

The OCR block recognizes the existing number in the detected ROIs. We perform some preprocessing prior to feeding the detected MSER regions to the OCR block. This preprocessing includes global binarization of the ROIs and removing small/large regions. We also perform post processing on the output of OCR engine as follows. Due to the specific font of our experiment numbers, as seen in Fig. 2c, digit 1 is similar to a vertical line segment (i.e. I) as opposed to standard digit 1 structure. So, the OCR block recognizes it as letter i. So, we replace each recognized i or I character by a digit 1. Next, we perform matching against the database of room numbers in the floor map and replace the detected number with the nearest existing number in the environment. Fig. 2c shows the result of the number recognition.

Having done text/number detection and localization, a rough estimate of the user's location is provided since the locations of text/number plates are known using the building floor plan. In order to refine the location estimate, we take advantage of SIFT [13] features of the image using [14]. We utilize a landmark-based method for fine localization similar to the supervised logo-based method discussed in [8]. To find the 3D coordinates of any query features, we use 3D

coordinates of the plate corners, which are measured in the training phase in advance.

2.3. Fine localization using Landmark-based methods

As stated, we extract SIFT features of the query image and utilize a landmark-based method to refine the location estimate. Landmark-based localization methods generally have two phases, training and test. In the training phase, a number of plate images (or only their SIFT features) are captured and stored in the database. This is necessary since in matching the query image against the image database, we usually encounter a huge view angle difference between query and database images. So, we basically have to perform wide baseline matching. View point invariance of SIFT features is limited. Therefore, in order to have a successful feature matching, which is crucial in success of fine localization, we should consider a number of images located at different view points for each plate. In our experiments, for each number plate, we captured 3 images located at a distance of 4 meters from the plate plane and separated by 1 meter horizontally. As experimented, this would result in a successful SIFT feature matching for all the query locations located in a hallway and in a field of view of more than 70 degrees of the number plate and can effectively compensate for the limited view point invariance of SIFT features .

In addition to storing the database images (features), we also need to measure and store the 3D (world) coordinates of at least four points in these images. This is the minimum number of points required to solve a perspective n-point (PnP) problem [15]. In PnP problem, the pose (position + rotation matrix) of a calibrated camera is computed from n 3D-2D point correspondences. We perform this by measuring the 3D coordinates of the plate corners similar to [8].

In the test phase, the user takes an image at its location and sends it to the server. In the server side, after performing text/number detection and recognition as explained, the SIFT features of the query image are extracted and matched against features of the corresponding database image (i.e. database image containing the same text/number). The details of this stereo feature matching are explained in the following.

2.3.1. *Stereo feature matching block*

Stereo feature matching between query and database images is performed to provide 3D labels for the query features. As we will explain later, most of the features detected in the database image are coplanar and their 3D coordinates can be computed using 3D coordinates of the number/text plate corners. Hence, we can label any query features matched to one of the coplanar database features.

In order to improve the stereo feature matching, we can benefit from the number/text plate location information inside the image provided by the OCR block as follows. First, we extract all the SIFT features from the entire query image. Then, we only select n-closest features to center of the number box in the image (depicted in Fig. 2c). n is a parameter of our system. These points are coplanar with the text/number plate with a high probability. The co-planarity is desired to compute the 3D coordinates of the query features, which is required to solve the PnP problem. We will explain this in details later. Since query feature points located close to the plate are supposed to be matched with the database ones located near the same plate, we can confine the database features to the set of n features closest to the number box plate. This would increase the matching accuracy and prevents matching to outliers. Fig. 3 shows the selected matched features in the query (left) and the database (right) images when $n = 15$.

Fig. 3: Feature matching for the 15-closest features to the number box center; all of matched feature are almost coplanar

2.3.2. *PnP block*

Text/number plate features are located on a single plain in the 3D space, i.e. the plane containing the brown door in Fig. 3. The way we select the n-closest features maximizes the probability of being on a single plain as explained. Co-planarity of the matched feature has the main advantage of easing the query feature points 3D labeling. We will explain it later.

When the corresponding database 2D (image) and 3D points are coplanar, there exist a homography relation between them [7]. As stated, for each database image, we measure the 3D coordinates of the plate corners ($\mathbf{X}_i$). We also manually determine and store their corresponding image coordinates ($\mathbf{x}_i^d$) in the database. If we represent these coordinates with $\mathbf{X}_i^d = \begin{pmatrix} X_i^d & Y_i^d & 0 & \lambda_i^d \end{pmatrix}^T$ and $\mathbf{x}_i^d = \begin{pmatrix} x_i^d & y_i^d & \gamma_i^d \end{pmatrix}^T$, respectively, we get

$$\mathbf{x}_i^d = \mathbf{P}_d \, \mathbf{X}_i^d \qquad \forall i \tag{1}$$

where $\mathbf{P}_d$ is the database camera matrix. Since the z coordinate of $\mathbf{X}_i^d$s is zero, we can define the shortened coordinates as $\tilde{\mathbf{X}}_i^d = \begin{pmatrix} X_i^d & Y_i^d & \lambda_i^d \end{pmatrix}^T$ similar to [7] and get

$$\tilde{\mathbf{x}}_i^d = \mathbf{H}_d \, \tilde{\mathbf{X}}_i^d \qquad \forall i \tag{2}$$

where $\tilde{\mathbf{x}}_i^d = K_d^{-1} \mathbf{x}_i^d$ is the normalized image coordinates of the database feature and K_d is the known calibration matrix of the database camera. $\mathbf{H}_d$ is a 3×3 invertible homography matrix that relates the 3D and image coordinates of the database image features.

We can use this homography to find the 3D coordinates of the query features ($\mathbf{x}_i^q$) in the test phase. Assume after stereo matching, database feature $\mathbf{x}_i^d$ is selected as the closest match. The shortened 3D coordinates of the query feature ($\tilde{\mathbf{X}}_i^q$) can be computed as

$$\tilde{\mathbf{X}}_i^q = \mathbf{H}_d^{-1} \, \tilde{\mathbf{x}}_i^d \qquad \forall i \tag{3}$$

where $\tilde{\mathbf{X}}_i^q = \begin{pmatrix} X_i^q & Y_i^q & \lambda_i^q \end{pmatrix}^T$. Once 3D and image coordinates of the query image as well as query camera calibration matrix are known, a PnP problem can be solved to find the complete pose (i.e. rotation matrix + location) of the query camera/user.

As explained, we have to find the $\mathbf{H}_d$ for each database image in the training phase. We convert this problem to a PnP problem and solve it with the robust method proposed by [16], which is also in the case of coplanar points. We found through experiments the method

proposed in [16], called RPnP, provides very good results even for the 4 noisy corners that we select manually in each image. For the database images, RPnP can solve for the rotation matrix ($\mathbf{R}_d$) and the translation vector ($\mathbf{t}_d$) in the following equation

$$\tilde{\mathbf{x}}_i^d = \mathbf{R}_d \tilde{\mathbf{X}}_i^d + \mathbf{t}_d \qquad \forall i \tag{4}$$

where $\tilde{\mathbf{x}}_i^d$ and $\tilde{\mathbf{X}}_i^d$ are the normalized image and shortened 3D coordinates of the database camera similar to our previous definitions. Comparing this with the equation considered in [8]

$$\tilde{\mathbf{x}}_i^d = \mathbf{H}_d \, \tilde{\mathbf{X}}_i^d \quad \forall i \tag{5}$$

and with some manipulation, one can find $\mathbf{H}_d$ from $\mathbf{R}_d$ and $\mathbf{t}_d$ as

$$\mathbf{H}_d = [\mathbf{R}_d^1 \quad \mathbf{R}_d^2 \quad \mathbf{t}_d] \tag{6}$$

where $\mathbf{R}_d^k$ is the k^{th} column of $\mathbf{R}_d$. This is the RPnP problem that should be solved in the training phase to find $\mathbf{H}_d$. The goal is essentially the opposite of [7] has targeted. Since, we have the complete pose information and want to get the homography matrix in order to label the query features. Another RPnP problem should be solved in the test phase to find a fine estimate of the user's location. Since we obtained $\tilde{\mathbf{X}}_i^q$ using (3) and the normalized image coordinates of the query features ($\tilde{\mathbf{x}}_i^q = K_q^{-1}\mathbf{x}_i^q$) using stereo matching, we can solve a PnP problem for

$$\tilde{\mathbf{x}}_i^q = \mathbf{R}_q \tilde{\mathbf{X}}_i^q + \mathbf{t}_q \qquad \forall i \tag{7}$$

and find $\mathbf{R}_q$ and $\mathbf{t}_q$. Here RANSAC should be combined with RPnP to make it robust to outliers (bad matches). After that, the user's location ($\mathbf{C_q}$) can be calculated as $\mathbf{C_q} = -\mathbf{R}_q^{-1}\,\mathbf{t}_q$.

2.3.3. Location estimation block

As stated, for each text/number plate, we store 3 images in the database. In the test phase, we compute three location estimates using each of these images. In the final location estimation block, we should combine these estimates from different database images. Thus, we linearly combine each location estimate $\mathbf{C}_q^i$ with weight w_i as

$$\mathbf{C}_q = \sum_{i=1}^{3} w_i \, \mathbf{C}_q^i \tag{8}$$

Since different database images are located at different view points from the plate, usually one or two of them give us appropriate number of matches and should be relied on for localization. So, we choose the weights to be the exponential of the RANSAC-verified matches to give more weights to the greater number of matches. That is, if the number of verified matches is N_i, then $w_i = e^{N_i}$.

3. EXPERIMENTAL RESULTS

As stated, [9] uses OCR for indoor localization. It has suggested using OCR when magnetic tracking fails and considered a localization error of 5 meters to be acceptable. Basically, it is considering the text/number plate location as a representative of the user's rough location. The authors have mentioned that whenever they need OCR-based localization in their system, they update the user's location on the map based on the seen characters in the query image. Since, they have not mentioned how this update exactly happens and we need to compare its performance with our *fine localization* system, we have to define a new benchmark. This benchmark would also be the representative of all methods that perform OCR-based rough localization. In the benchmark, we assign the centroid of all test locations as the estimated rough location. Centroid is the point/location that has the minimum average of square distances (errors) to all the test locations. Hence, we consider the minimum average error case among the OCR-based rough localizers as the benchmark.

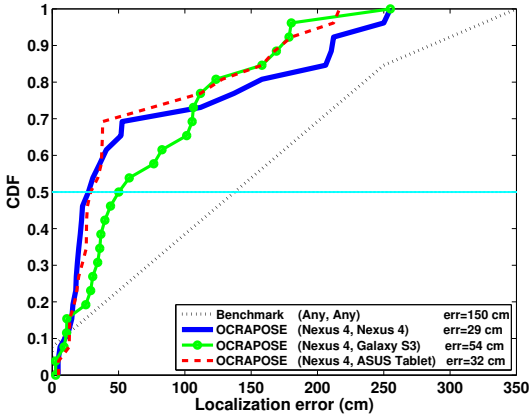


Fig. 4: CDF of the localization error (cm) for benchmark and OCRAPOSE using different test devices. Median error of each scenario is mentioned in the legend

We perform camera calibration in advance as calibration matrices are assumed to be known. OCR function deployed in MATLAB R2014a was utilized for number recognition. MATLAB has incorporated Google Tesseract engine [17], which uses convolutional neural network for character recognition. We stored 3 images in the database as explained. Due to limited space available, we only show the results of one scenario. In the scenario, we have considered 26 locations located in a hallway in front of the number plate. These points are located on a $7m \times 1m$ rectangular grid. The grid is located in a hallway at a minimum distance of 1 meter from the plate wall. Points are separated by 1m horizontally. Each consequent points on left or right wings of the frame are separated by 50 cm. This results in a total number of 26. Fig. 4 shows the CDF of the localization error for the proposed system (OCRAPOSE) and the benchmark. OCRAPOSE performance was tested in three scenarios with devices shown in the legend as pairs. The pair show the training and test devices, respectively. Test was performed for Google Nexus 4, Samsung Galaxy S3 and ASUS Transformer tablet while Nexus was used for training. These are all commercial cellphone or tablets and both training and test were done by simply holding them towards the number plate with an arbitrary orientation. As seen in the figure (legend), the median error in the benchmark is 150 cm, while it is below 54 cm for any test devices using OCRAPOSE. So, OCRAPOSE has refined the location by at least a factor of 3 approximately. As seen, testing with a different device weakly affects the accuracy provided the calibration matrix of devices are known.

4. CONCLUSION

We propose an image-based system, called OCRAPOSE, applicable for localization in office buildings, chain stores, airports, etc. The proposed system combines OCR and landmark-based techniques to provide a fine location estimate. It was demonstrated that OCRAPOSE achieves a median localization error of less than 50 cm for test positions located up to 7 meters away from a $20cm \times 30cm$ number plate using different test devices in a university building scenario.

5. REFERENCES

[1] Paula Tarrío, Juan A Besada, and José R Casar, "Fusion of rss and inertial measurements for calibration-free indoor pedestrian tracking," in *Information Fusion (FUSION), 2013 16th International Conference on*. IEEE, 2013, pp. 1458–1464.

[2] Mohamed Maher Atia, Michael Korenberg, and Aboelmagd Noureldin, "A wifi-aided reduced inertial sensors-based navigation system with fast embedded implementation of particle filtering," in *Mechatronics and its Applications (ISMA), 2012 8th International Symposium on*. IEEE, 2012, pp. 1–5.

[3] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "Tumindoor: An extensive image and point cloud dataset for visual indoor localization and mapping," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, Sept 2012, pp. 1773–1776.

[4] Jiebo Luo, Dhiraj Joshi, Jie Yu, and Andrew Gallagher, "Geotagging in multimedia and computer vision? a survey," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 187–211, 2011.

[5] Jason Zhi Liang, Nicholas Corso, Eric Turner, and Avideh Zakhor, "Image based localization in indoor environments," in *Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on*. IEEE, 2013, pp. 70–75.

[6] Hamed Sadeghi, Shahrokh Valaee, and Shahram Shirani, "A weighted knn epipolar geometry-based approach for vision-based indoor localization using smartphone cameras," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014 IEEE 8th*. IEEE, 2014, pp. 37–40.

[7] Yang Yang, Qixin Cao, Charles Lo, and Zhen Zhang, "Pose estimation based on four coplanar point correspondences," in *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*. IEEE, 2009, vol. 5, pp. 410–414.

[8] Hamed Sadeghi, Shahrokh Valaee, and Shahram Shirani, "Semi-supervised logo-based indoor localization using smartphone cameras," in IEEE PIMRC, 2014.

[9] Jason Orlosky, Takumi Toyama, Daniel Sonntag, Andras Sarkany, and Andras Lorincz, "On-body multi-input indoor localization for dynamic emergency scenarios: fusion of magnetic tracking and optical character recognition with mixed-reality display," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*. IEEE, 2014, pp. 320–325.

[10] Jonathan Harel, Christof Koch, and Pietro Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, 2006, pp. 545–552.

[11] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[12] Arpit Jain, Xujun Peng, Xiaodan Zhuang, Pradeep Natarajan, and Huaigu Cao, "Text detection and recognition in natural scenes and consumer videos," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1245–1249.

[13] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[15] Radu Horaud, Bernard Conio, Olivier Leboulleux, and Bernard Lacolle, "An analytic solution for the perspective 4-point problem," in *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on*. IEEE, 1989, pp. 500–507.

[16] Shiqi Li, Chi Xu, and Ming Xie, "A robust o (n) solution to the perspective-n-point problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1444–1450, 2012.

[17] Ray Smith, "An overview of the tesseract ocr engine.," in *ICDAR*, 2007, vol. 7, pp. 629–633.