EFFICIENT INDOOR LOCALIZATION VIA REINFORCEMENT LEARNING

Dimitris Milioris

Nokia Bell Labs, 91620 Nozay, France
Massachusetts Institute of Technology, 77 Massachusetts Avenue, MA 02139, USA
Email: milioris@mit.edu; dimitrios.milioris@nokia-bell-labs.com

ABSTRACT

In recent years the widespread use of Global Positioning Systems (GPS) has enhanced our abilities to find our way in outdoor environments, dispensing our over-reliance on visual features. However, GPS technologies do not solve the problem of localization and way finding in indoor environments due to transient phenomena such obstacles and fading. With the development of computer vision and deep learning techniques, vision-based localization and way finding have recently received special attention in the scientific community. In this paper we propose a method which learns to localize with high accuracy while minimizing the total number of processes based on reinforcement learning. We train our model on a dataset at MIT campus and we evaluate its performance by comparing with state-of-the-art techniques, obtaining higher accurate results.

Index Terms— Deep Learning, Reinforcement Learning, Bayesian Filtering, Localization and Navigation, Space Visual Similarity

1. INTRODUCTION

Humans heavily rely on visual cues to position themselves and find their way in space. Indeed, visual diversity has for long been considered essential in building the character of a place, helping people to locate themselves in a physical space, and even improving the correlation of restorative experiences of spaces [1]. Thus, knowing to identify and correlate spatial visual cues is a critical asset of way finding.

In recent years the widespread use of Global Positioning Systems (GPS) has enhanced our abilities to find our way in outdoor environments, dispensing our over-reliance on visual features. However, GPS technologies do not solve the problem of way finding in indoor environments for several reasons, including non-line-of-sight conditions and relatively smaller-scale spaces requiring spatial decisions to be taken with higher accuracy [2, 3, 4].

Solutions to address indoor localization and way finding include deploying WiFi and Bluetooth beacons, using RFID reader antennas combined with active or passive transceivers [5], placing visual landmarks in known positions that can be visually recognized by cameras, or using personal cameras and comparing the pictures with image databases [6, 7, 8].

With the development of computer vision and deep learning techniques, vision-based localization and way finding have received important attention in the scientific community. On the one hand, vision-based applications have showed the potential to enhance the ability of people to recognize spatial features and locate oneself indoors, even for visually-impaired people [9]; and on the other hand scholars have been trying to improve the vision-based accuracy and at the same time circumvent the high costs of these techniques some involving expensive specialist hardwares not suited to mobile devices [10].

An important part of vision-based localization techniques is found in the robotics and artificial intelligence literature, which rely on the distribution of landmarks in space that can be identified by cameras installed on robots. Other techniques are based on matching the photos taken by the robots with datasets of tagged images of the same environment. Although effective for robots, one caveat to transporting this technique to humans is that such robots need to carry computing devices, which would be impractical for humans [11].

Another caveat is that unless the landmark is in the image, the space is not identified. So, in our case the goal is not simply find the most efficient system to locate someone is space, but propose a system that regain the human capability of locating oneself in space by reading visual cues and spatial features and matching such inputs with previous visual experiences. In the case of computer-vision techniques, this visual experience of space takes the form of large image datasets. Previous research has been addressing this issue using Convolution Neural Networks (CNN) to solve object classification and object detection problems [12], as well as image recognition, location and directionality [13]. ResNet [14] showed great performance in multi-layer CNNs for object detection and face recognition, but it is a heavy procedure for simple image similarity detection.

The question that interests us is how to use computer vision and machine learning techniques to read a space as we

Dr. Dimitris Milioris would like to thank the MIT SCL for providing the dataset and supporting this research.

humans do, using an egocentric visual perception based on visual cues selected while navigating space. In order to do that, we use thousands of photos extracted from numerous videos to train our model, and then try to identify the physical space from a single image taken at each timestep. We compare our method with other similar vision-based techniques designed to accurately find oneself in space using ordinary mobile devices.

The remaining part of this paper is organized as follows: Section 2 gives the motivation for this work, while Section 3 presents the architecture based on a deep neural network and Bayesian filtering for indoor navigation. Section 4 presents the data collection at the MIT campus and the experimental results. Finally, we give conclusions and future work in Section 5.

2. MOTIVATION

Localization and way-finding techniques using ego-centric images have been gaining momentum with the development of machine learning techniques, and the availability of low-cost, and portable computer [15]. Machine learning techniques and first-person-view images for both localization and way finding was proposed by [16]. Users wear a portable camera and receive audio instructions from what they call 'virtual usher' to find their destinations in buildings never visited before. Our paper also addresses the problem of way finding in indoor environments using computer vision and machine learning techniques, but adds up a particular challenge: the test site has very little distinctive features.

Such site selection intends to mimic situations when people are not used to a particular indoor environment and needs to locate themselves fast and accurately to make timely decisions. Finding such solution has notable relevance in several situations, such as with emergency crews navigating unfamiliar environments: they need to find themselves in unknown spaces in real time. Once this localization is achieved, it is possible to connect this position with existing maps of the environment within a way-finding system that suggests the shortest or safest route to the emergency situation.

Rather than testing our model in a space with remarkable visual features, we chose to test it in a bland space: the hundred-meters long "infinite corridor" at MIT campus, and indoor environment with few visually-distinctive cues. Another motivation for this research was to test our methodology based on images taken with a trivial smart phone camera, not requiring any specific hardwares.

3. PROPOSED SYSTEM ARCHITECTURE

In order to design a model to estimate a dynamic system's state using knowledge from the scenes around the user, we prefer to use Bayesian filtering. Assume that in time t we have the following random variables: y_t is the state of the user, which includes its (x, y) coordinates and its orientation; o_t is the observation received by the system, and a_t is the action taken at time t. The belief $B(y_t) = prob(y_t|o_{1:t-1}, a_{1:t-1})$ is the probability distribution over y_t conditional to the past observations and actions, and under Markov assumptions it can be computed by:

$$B^*(y_t) = \sum_{y_{t-1}} prob(y_t|y_{t-1}, a_{t-1})B(y_{t-1}) \quad (1)$$

$$B(y_t) = \frac{1}{Z}L(o_t)B^*(y_t) \quad (2)$$

with $L(o_t)$ being the $prob(o_t|y_t)$ which is the likelihood of getting o_t while the actual location is y_t and Z being the normalization constant as defined in Bayesian filtering. $B(y_0)$ can be defined based on our prior location estimation of the user, however in real time, since we do not know the user's location, we have to assume it is uniformly adjusted over all possible locations.

In order to help the system initiate the process, we give as an input the map design D, therefore our goal is to estimate the $B(y_t) = prob(y_t|o_{1:t}, a_{1:t-1}, D)$ by learning a policy $\pi(a_t|B(y_t))$.

Driven by the work in [17], we use a grid-based representation of Belief B which offers structure advantages over other representations. In our work, B is represented as a 3-dim tensor *(orientation, x axis, y axis)*, where the (k, x, y) element gives B of being on the right state. The same occurs for the likelihood tensor which describes the observation being in a state given the user's state.

Two main components define the architecture of the proposed system:
(a) the perceptual model, which has as an input the user's observation o_t and gives as an output the tensor likelihood $L(o_t)$. Belief, after observing o_t is given by the following equation

$$B(y_t) = \frac{1}{Z}B^*(y_t) \odot L(o_t) \quad (3)$$

where $\odot$ is the Hadamard product. The feature representation of each observation is obtained by using a deep convolutional network described in Section 3.1.

(b) the policy model, which has as an input the $B(y_t)$ and as an output the probability $\pi(a_t, B(y_t))$ and defines the next action. The Belief tensor is updated at each location based on the action taken by the user and based on the transition function T_f [18]. More specifically, left and right directions-actions are moved by the transition function. The same goes for the forward direction-action, unless we have an obstacle. The policy model is trained using reinforcement learning, which connects the Belief with the map design and is given as an input to the deep convolutional network to generate

the policy prediction. The description follows in Section 3.2. At each time t the user gets a reward which is equal to the maximum probability of located in any state, and that's how the entropy of the Belief tensor is reduced and we get a more efficient localization. Correct prediction $y^* = max_{y_t} B(y_t)$ gives 1 as a positive reward to the user. The goal for every timestep t is to calculate the sum of rewards and maximize that based on learning a policy $\pi(\alpha|o)$. The system architecture, for a given time t is shown in Fig. 1.

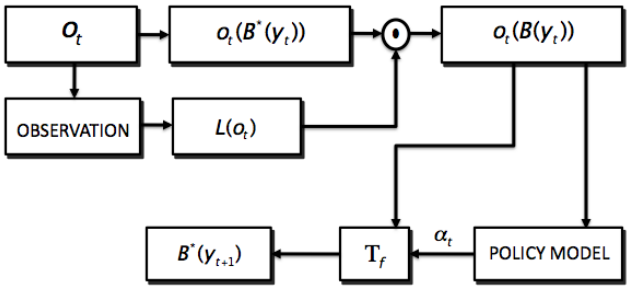

Fig. 1. Proposed System Architecture.

3.1. Deep Convolutional Network - Perceptual model

The proposed deep neural network, as shown in Fig. 2, consists of five layers; the first three are convolution layers, followed by a flatten layer, which reshapes the tensors. Then, a fully connected layer follows in order to learn the non-linear relations and extract the features. The model is built on TensorFlow [19], and used 2x2 pooling with strides set to 1 in all dimensions. Biases are added to the results of the convolution layers, with a bias-value added to each filter-channel.

The Rectified Linear Unit (ReLU) [20] is used in order to add some non-linearity to the formula and gives us insights to learn more complicated functions. An approximation to the rectifier is the analytic function $f(x) = \log(1+\exp x)$, where x is the input to a neuron, and $f^{'}(x)$ is the logistic function It is applied extensively in computer vision using deep neural networks. The ReLU is executed before the pooling process, but we save time and cost of the ReLU operations when we perform max-pooling first.

The first convolution layer gets as input an image and down sample that image by using 2x2 max-pooling. The same process follows for the second and third convolution layer. Then the output of the third convolution layer is fed to the fully connected network. The ReLU is used to learn the non-linear relations in the fully connected layer.

Global average pooling [21] has been proved to regularize structures, therefore avoids overfitting when compared to the fully connected layers introduced to the model. We use the global average pooling to implement the Class Activation Mapping (CAM), because it enables the function to distinguish between different regions of interest within the images, and get relevant information about the features extracted in a single forward process [22].

We define as $A_m(i,j)$ the last convolution layer's feature map m, w_c as the weight vector between the global average pooling and the softmax function of the predicted class c. In this case, w_c is the optimal linear combination weights for the feature map $A_m(i,j)$. We can therefore note that the matrix Ω_c of the last convolution layer can be written as

$$\Omega_c = \sum_m w_c A_m(i,j)$$

and by performing up sampling, we get a synthesis of the input image, and sense the visual spatial feature of the model, such as style and pattern of the physical space.

TensorFlow gets all the info described above and runs the Adam Optimizer [23], which is an advanced form of Gradient Descent. Gradient Descent maintains a single learning rate for all weight updates which does not change during training. The Adam Optimizer computes individual adaptive learning rates for each network weight from estimates of first and second moments of the gradients. It basically combines the advantages of (a) the Adaptive Gradient Algorithm, which maintains a per-parameter learning rate that improves performance on problems with sparse gradients, such as computer vision problems, and (b) the Root Mean Square Propagation, which captures the quick change of gradients (average of recent magnitudes), and works well for non-stationary problems, such as noisy data.

3.2. Reinforcement Learning - Policy model

During the policy phase, the Map design along with the Belief tensor are fed into the network. Then, two convolutional layers of 7x7 with a stride of 3 are connected, followed by a flatten layer which is an input to a fully connected layer. The last five actions and the time t are added in order to prevent the prediction from being stuck into left and right directions and take a decision at each episode. The history actions along with time t are fed into an embedding layer, which contacts with the fully-connected layer's output. The generated vector is fed to a fully-connected layer to extract the policy and the value function. The proposed system is shown in Fig. 3 and is aligned with the work in [24].

4. DATA COLLECTION AND EXPERIMENTAL RESULTS

MIT campus consists of several buildings, tunnels, corridors and rooms, which have slightly different architecture. Newcomers, visitors, guests and tourists, usually face difficulty

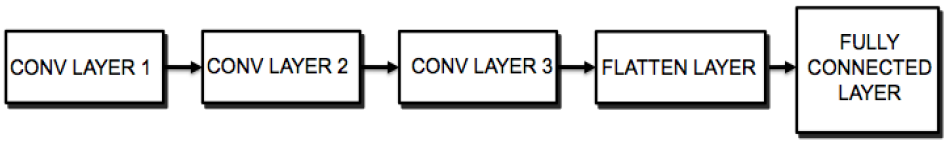

Fig. 2. Proposed deep learning network for Perceptual model.

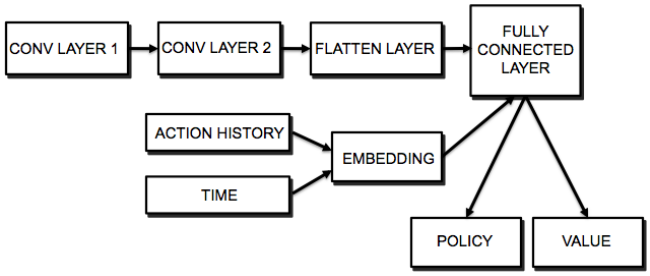

Fig. 3. Proposed reinforcement learning network for Policy model.

in finding their way when navigating within the campus. In order to address the problem of way finding through our deep learning model we were based on the MIT's floor plans. Basically, we divided the twenty six buildings of the east campus into thirty five segments of interest which range from 30 to 500 m^2. Then, large segments were divided into smaller room level areas in order to increase the spatial resolution of the physical space. More specifically, lobby 7 was divided into an area of ten segments: four, three and three segments for the 1st, 2nd and 3rd floor respectively. A more detailed description of the dataset can be found in [25].

The videos were captured using an action camera to have a more polished dataset. Due to the change and dynamics of human movement, obstacles and variance of the light conditions, we collected our dataset in different times of the day as well as in different angles, height, and routes in order to collect more complete data. The 1700 testing images were captured by an iPhone 6s sequentially in random locations and random time of the day. The resolution of the videos was set at 720p, 30fps, and we collected 102 videos in total of arbitrary length. We extracted the images from these videos and rescale them in 256x256 by running an *ffmpeg* command in order to remove blur and get better quality images. That resulted to approximately 600,000 images. We then randomly split the dataset into 80% for training and 20% for validation and labeled the data by marking the actual location of each image. The proposed deep learning network for the perceptual model was implemented on TensorFlow. The model was trained on Amazon Web Service in a parallel computing environment which consists of 3584 CUDA cores and 16GB of RAM, and took less than four hours.

Along with our proposed system (perceptual model), we evaluated the performance of AlexNet [12], NIN [21], and ResNet [14] and DeepSpace [25]. During the training and validation, all networks achieved similar results. However, during the testing dataset, our proposed system performs better and gives an accuracy of 90%, which is higher than the other models. The experimental results along with a more detailed description of the model can be found in [25]. The human brain identifies interior places by recognizing and combining objects, colors, patterns and orientation. In order to build a model which shows the difference of interior places, we need to place together a multi-object recognition which combines different filters.

A different evaluation metric for the accuracy is the ratio of the episodes of the correct localization over a thousand episodes. The state-of-the-art algorithms used to compare are: (a) Markov Localization and Active Markov Localization [26]. These algorithms both use the Resnet-18 [14] model to extract the feature representations for the user observations. Table 1 shows that the proposed system outperforms the Markov methods in terms of episodes accuracy.

Table 1. Performance Evaluation: Markov Localization, Active Markov Localization, and our proposed system

Model	With CAM	Without CAM
Markov Localization	0.617	0.351
Active Markov Localization	0.671	0.365
Proposed System	**0.951**	**0.512**

5. CONCLUSIONS

In this paper we proposed an indoor space recognition method using neural networks on images extracted from videos captured at MIT campus. We discussed a model based on Bayesian filtering along with perceptual and policy components based on deep and reinforcement learning respectively. Experimental results showed that the proposed model learns and recognizes the interior space with high accuracy and helps people navigate. As a future work, we plan to work on the recognition of corners and angles by designing extra layers, and understand the correlation between confusion degree and spatial features of indoor environments.

6. REFERENCES

[1] P. J.Lindal, T. Hartig,"Architectural variation, building height, and the restorative quality of urban residential streetscapes", in *Journal of Environmental Psychology*, Vol. 33, pp. 26–26, 2013.

[2] L. Mainetti, L. Patrono, I. Sergi, "A survey on indoor positioning systems", in *22nd International Conference on Software, Telecommunications and Computer Networks* 2014, (DOI: 10.1109/SOFTCOM.2014.7039067).

[3] P. Mirowski *et. al*, "Probabilistic RF Fingerprinting and Localization on the Run", in *Bell Laboratories Technical Journal, Issue on Data Analytics*, Vol. 18, No. 4, 2014.

[4] D. Milioris *et al.*, "Low-dimensional signal-strength fingerprint-based positioning in wireless LANs", in *Ad Hoc Networks Journal, Elsevier*, Vol. 12, pp. 100–114, Jan. 2014.

[5] D. Milioris, M. Bradonjic and P. Muhlethaler, "Building complete training maps for indoor location estimation", in *Proc. IEEE INFOCOM*, pp. 75–76, 2015.

[6] J. Xiao, Z. Zhou, Y. Yi, L. Ni, "A Survey on Wireless Indoor Localization from the Device Perspective", in *ACM Computing Surveys (CSUR)*, Vol. 46, Issue 6, pp. 1–31, 2016 (DOI:10.1145/2933232).

[7] N. Fallah, I. Apostolopoulos, K. Bekris, E. Folmer, "Indoor Human Navigation Systems: A Survey", *Interacting with Computers*, Vol. 25, Issue 1, pp. 21–33, 2013 (DOI:10.1093/iwc/iws010)

[8] S. A. Cheraghi, V. Namboodiri, L. Walker, "GuideBeacon: Beacon-based indoor way finding for the blind, visually impaired, and disoriented", in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Kona, HI, USA, March 13-17, 2017, (DOI:10.1109/PERCOM.2017.7917858).

[9] D. Zhang, D-J. Lee, B. Taylor, "Seeing Eye Phone: a smart phone-based indoor localization and guidance system for the visually impaired", in *Machine Vision and Applications*, Vol. 25, Issue 3, pp. 811–822, 2014, (DOI:10.1007/s00138-013-0575-0).

[10] R. Clark, N. Trigoni, A. Markham, "Robust vision-based indoor localization", in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pp. 378–379, Seattle, WA, USA, April 14-16, 2015, (DOI:10.1145/2737095.2742929).

[11] L. A. Guerrero, F. Vasquez, S. F. Ochoa, "An Indoor Navigation System for the Visually Impaired", in *Sensors*, Vol. 12, Issue 6, pp. 8236–8258, 2012, (DOI:10.3390/s120608236).

[12] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks". Advances in neural information processing systems, 2012, pp. 1097-1105.

[13] G. Ni, R. Yu, D. Zhao, T. Wu, B. Dai, "Vision-based localization in campus with a multi-stage framework", in *2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 58-61, 2017, (DOI:10.1109/ACIRS.2017.7986065).

[14] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[15] A. Betancourt, P. Morerio, C. S. Regazzoni, M. Rauterberg, "The Evolution of First Person Vision Methods: A Survey", in *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 25, Issue 5, pp. 744–760, 2015, (DOI:10.1109/TCSVT.2015.2409731).

[16] L. Li, Q. Xu, V. Chandrasekhar, J.-H. Lim, C. Tan, M. A. Mukawa, "A Wearable Virtual Usher for Vision-Based Cognitive Indoor Navigation", in *IEEE Transactions on Cybernetics*, Vol. 47, Issue 4, pp. 841–854, 2017, (DOI:10.1109/TCYB.2016.2530407).

[17] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian Filtering for Location Estimation", in *IEEE Pervasive Computing*, 2(3):24–33, 2003.

[18] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation", (arXiv:1702.03920), 2017.

[19] "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", Software available in `https://www.tensorflow.org/`

[20] R. Hahnloser, R. Sarpeshkar, M A Mahowald, R. J. Douglas, and H.S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit", in *Nature*, 405, pp. 947–951, 2000, (DOI:10.1038/35016072).

[21] M. Lin, Q. Chen and S. Yan, "Network In Network", 2013, arXiv:1312.4400

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 27-30, 2016, (DOI:10.1109/CVPR.2016.319).

[23] K. Diederik and J. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference for Learning Representations*, San Diego, 2015.

[24] V. Mnih, A. Puigdomenech Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning", in *International Conference on Machine Learning*, pp. 1928–1937, 2016.

[25] F. Zhang, F. Duarte, R. Ma, D. Milioris, H. Lin, C. Ratti, "Indoor Space Recognition using Deep Convolutional Neural Network: a case study at MIT Campus", Oct. 2016, (arXiv:1610.02414).

[26] D. Fox, W. Burgard, and S. Thrun, "Active Markov Localization for Mobile Robots", in *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.