Where is my Phone?

Personal Object Retrieval from Egocentric Images

Cristian Reyes  
Insight Center for Data Analytics  
Dublin, Ireland  
cristian.reyes@estudiant.upc.edu

Eva Mohedano  
Insight Center for Data Analytics  
Dublin, Ireland  
eva.mohedano@insight-centre.org

Kevin McGuinness  
Insight Center for Data Analytics  
Dublin, Ireland  
kevin.mcguinness@dcu.ie

Noel E. O'Connor  
Insight Center for Data Analytics  
Dublin, Ireland  
noel.oconnor@dcu.ie

Xavier Giro-i-Nieto  
Universitat Politecnica de Catalunya  
Barcelona, Catalonia/Spain  
xavier.giro@upc.edu

ABSTRACT

This work presents a retrieval pipeline and evaluation scheme for the problem of finding the last appearance of personal objects in a large dataset of images captured from a wearable camera. Each personal object is modelled by a small set of images that define a query for a visual search engine. The retrieved results are reranked considering the temporal timestamps of the images to increase the relevance of the later detections. Finally, a temporal interleaving of the results is introduced for robustness against false detections. The Mean Reciprocal Rank is proposed as a metric to evaluate this problem. This application could help into developing personal assistants capable of helping users when they do not remember where they left their personal belongings.

Keywords

Lifelogging; egocentric; retrieval; wearable camera

1. INTRODUCTION

The interest of users in having their lives digitally recorded has grown in recent years thanks to the advances on wearable sensors, egocentric cameras being among the most informative ones. Since wearable cameras are mounted on the user, they are ideal for gathering visual information from everyday interactions.

People interact with their personal belongings several times over the course of the day and, in many cases, they are unintentionally lost because users forget the last location where they were handled. Wearable cameras can help users to retrieve candidate locations to contain the query object. As the forward egocentric view of these cameras often captures the manipulations with personal belongings as well enough pixels from the background to quickly identify the location. Adopting a computer vision approach for finding lost objects is more versatile than a sensor-based one, where some sort of tracking device must be explicitly attached to each object. Cameras do not need any additional device nor an explicit registration of the object.

There has already been an extensive work on personal object detection in vision, especially when considering video surveillance cameras from CCTV [24]. Wearable cameras offer two main advantages with respect to solutions based on CCTV footage. Firstly, they move together with the user, so their range is not restricted to a specific location. Secondly, a single device can be highly resilient to the object occlusions, as the camera normally takes the same point of view of the user, while a CCTV system will require multiple cameras to cover all points of views. However, and similarly to CCTV-based systems, these capturing devices typically generate very large volumes of images daily, so finding the relevant images to solve the problem is not a simple task. An automatic, efficient and scalable retrieval system is necessary to address these situations.

In this work, we assess the potential of egocentric vision to help the user answer the question Where did I leave my …?. We address it as a time-sensitive image retrieval problem. We explore the design of a retrieval system for this purpose, focusing on the visual as well as on the temporal information. This application could help in developing personal assistants capable of helping users when they do not remember where they left their personal belongings.

This paper focuses in the problem of personal object retrieval from egocentric images with the following contributions:

- A reranking strategy based on temporal interleaving of those candidate images to contain the query object.
- A comparison between center bias and saliency maps for the spatial weighting of visual features in the target database.
- A proposal of the Mean Reciprocal Rank (MRR) as evaluation metric.

Explore the usefulness of visual saliency maps to improve the performance of the visual search engine.

The paper is structured as follows. Section 2 provides an overview of classic applications for lifelogging data together with previous works that have addressed the problem of finding lost objects with CCTV footage. Section 3 presents the proposed solution based on
egocentric images, which is based on an off-the-shelf visual search whose retrieved occurrences are reranked to introduce temporal diversity. Section 4 evaluates the proposed solution with the Mean Reciprocal Rank (MRR), which is proposed as the reference metric to evaluate personal object retrieval systems. Finally, Section 5 presents the conclusions and draws future research directions.

2. RELATED WORK

The use of wearable cameras to create persistent personal memories has been tightly associated to the concept of lifelogging [15]. Early applications of these devices have focused in healthcare [8, 14], with works exploring their applications for patients with mild dementia [10]. More recent works have also explored their applications in affective computing [16], social interactions [32], and activity recognition [22].

The detection of objects in visual lifelogs has been explored with different applications. A very popular one is dietary analysis based on the food captured by the camera [25, 2]. Human-object interactions have been recognized by combining object recognition, motion estimation, and semantic information [27], and also by using the hand-object interaction [12, 33]. Object recognition is not only useful for object-based detections but also for event identification using the object categories that appear in an image [19] or activity recognition based on the object’s frequency of use [31]. The identification of the active object in the scene was explored in [3] with the help of visual saliency models.

While object detection and recognition techniques are relevant for this work, in our application we address a retrieval problem, where a ranked list of images from database is shown to the user to help him/her locate their lost object. Previous works from lifelogging have defined retrieval problems for events [9], audio pieces [28], as well as summarization [5] and novelty-detection [1].

The problem of retrieval is not only solved with a search based on a similarity metric, as it often also requires the introduction of the notion of diversity. The text-based seminal work of Carbonell & Goldstain [4] recognizes that pure relevance ranking is not sufficient, so the authors proposed a reranking method that combines independent measurements of relevance and diversity into a single metric to maximize. In image retrieval field, diversification has also shown to increase user satisfaction in ranked results [30]. Diversity in social image retrieval was one of the focus of the MediaEval benchmarks [18] benchmarks and attracted the interest of many groups working in this field. In our problem of personal object retrieval, diversity is especially important because we are mainly interested in the location of the object, more than the object itself. That is, our system should not provide the higher qualities appearances of the object in the database but give hints to the user about the location where the object was last seen.

3. METHODOLOGY

Our goal is to rank the images captured by a wearable camera during a day based on their likelihood to depict the location of a personal object. In our problem we have defined the following sets of images as inputs to the system:

- The query set \( Q \): For each object to be retrieved, a set of exemplar images containing the object is necessary to define the query for the system.

- The target set \( I \): Dataset of images captured by the wearable camera. For each day, this set contains 2,000 images captured throughout the day.

![Figure 1: Global architecture of the pipeline based system.](image)

Figure 1 depicts the system architecture. It can be divided in two main blocks: a visually sensitive ranking one followed by a temporally sensitive one. The visual block is based on using a pretrained deep convolutional neural network to generate a local representations of an image, and encoding these usiag of bag of words aggregation [23]. At query time, each image in the target set is assigned a score representing its visual similarity with the query object. The temporal block is composed by a first step that selects candidate images and a second one which enhances the ranking using temporal information. We included configuration flags for each stage to determine the most appropriate set up.

3.1 Baseline

As far as we are aware, there is no previous work on finding lost objects in egocentric images. We decided to define as a baseline the simplest approach for the resolution of the problem: a simple temporal sorting of the images based on their time stamp, with the first image in the ranking being the last one taken by the camera. This mimics the case of a user sequentially browsing through the full sequence of images for a day in reverse order, the obvious course of action for someone seeking a lost personal item.

3.2 Ranking by visual similarity

The goal of this stage is to create a ranking \( R_o \) of the \( I \) set for a given \( Q \) set. The ranking is based only on the visual information of the images. We explored several configurations and variations of a Convolutional Neural Network Bag-of-Words (BoW) similarity model proposed in [23]. This model is based on the off-the-shelf features learned with the VGG16 network [29] trained on ImageNet [7], so no feature learning nor fine-tunning was applied.

The BoW is popular in the image retrieval community and is the basis for some of the best performing techniques of the TRECVID Instance Search Task 2015, where an object instance must be found in a large dataset of videos. Their easy indexing and implementation as an inverted file or multiplication of sparse matrixes makes it a very common approach for content-based object retrieval systems.

3.2.1 Encoding the query images

A function \( f : Q \rightarrow f(Q) \in \mathbb{R}^n \) aims at building a single query vector \( \vec{q} \) by gathering the information of all images in \( Q = \{q_1, q_2, \ldots, q_Q\} \) to obtain \( \vec{q} = f(Q) \). Three different approaches have been explored to define \( f \), illustrated in Figure 2:

- a) Full Image (FI): The \( \vec{q} \) vector is constructed by averaging the frequencies of the visual words of all the local CNN features from the query images.

- b) Hard Bounding Box (HBB): The \( \vec{q} \) vector is constructed by averaging frequencies of the visual words that fall inside a query
constructing and salient object detection [20]. This approach is inspired by previous works in the field of inversely weightens the features with the distance to the center of the vector is built using the visual words of all the local CNN features from the image. This approach is inspired by previous works in the field of salient object detection [20].

Figure 2: Examples of the different masking strategies applied in a query image. Left: Full Image, center: Hard Bounding Box, Right: Soft Bounding Box.

3.2.2 Encoding the target images

A similar procedure is applied to the set of target images I, the daily images in our problem. A function \( g : I \rightarrow \mathbb{R}^n \) is defined to build a feature vector \( \vec{i}_j = g(i_j) \) for each image \( i_j \in I \). Three different definitions of the \( g \) function have been studied (Figure 3):

a) Full Image (FI): The \( \vec{i}_j \) vector is built using the visual words of all the local CNN features from the \( i_j \) image.

b) Center Bias (CB): The \( \vec{i}_j \) vector is built using the visual words of all the local CNN features from the \( i_j \) image but it inversely weights the features with the distance to the center of the image. This approach is inspired by previous works in the field of salient object detection [20].

c) Saliency Mask (SM): The \( \vec{i}_j \) vector is built using the local CNN features of the whole image, but this time weighting their frequencies using a saliency map generated using a computational model of visual saliency. Using saliency maps for object detection and recognition has been previously proposed in [17, 11, 3].

Following [23], an assignment map is extracted from the \( conv-5 \). layer of the VGG-16 pre-trained convolutional neural network, giving a \( 32 \times 42 \) assignment map.

Saliency maps are calculated for each image using the pre-trained SalNet [26] CNN. This network produces maps that represent the probability of visual attention on an image, defined as the eye gaze fixation points. We downsample the saliency maps to match the size of the assignment maps by average pooling over local blocks. After downsampling a vector of weights \( w = (w_1, ..., w_{32 \times 42}) \) is constructed and \( L_2 \)-normalized.

Finally, each \( \vec{i}_j \) feature vector is compared with the \( \vec{q} \) query feature vector using cosine similarity \(^1\) between \( \vec{i} \) and \( \vec{q} \) and obtain the \( \nu \) score. Then the visual ranking \( R_v \) is produced by ordering the images in \( I \) according to their \( \nu \) score.

3.3 Detection of candidate moments

The visual ranking \( R_v \) provides an ordered list of the images based on their likelihood to contain the object. Notice that in our problem this information might not always be useful. The last appearance of the object does not need to be the most similar to the query in visual terms. Taking this into account we introduced a post-processing to the visual search ranking.

The first step in this post-processing is determining which of the images in the ranked list should be considered as likely to contain the query object. This is achieved by thresholding the list and considering as candidate images (\( C \)) those with scores higher than the threshold, and discarded images (\( D \)) those with scores below it.

Two different thresholding techniques were considered in order to create the \( C \) and \( D = I \setminus C \) sets.

a) Threshold on Visual Similarity Scores (TVSS): This technique consists in building the set of the candidate images as \( C = \{ p \in I : \nu_p > \nu_{th} \} \), where \( \nu_{th} \) is a learned threshold. It is an absolute threshold that the visual scores have to overcome to be considered as candidates.

b) Nearest Neighbor Distance Ratio (NDRR): This strategy is inspired by Lowe [21]. Let \( \nu_1 \) and \( \nu_2 \) be the two best scores in the ranked list, then the candidates set is defined as

\[
C = \left\{ i \in I : \frac{\nu_1}{\nu_2} > p_{th} \right\}.
\]

In this case, it is an adaptive technique that sets the threshold depending on the ratio of the scores of two best visually ranked images.

Both techniques require to set either \( \nu_{th} \) or \( p_{th} \). These values cannot be chosen arbitrarily, so they were learned from a training process, described in section 4.3.

3.4 Temporally aware reranking

Once candidate images have been selected based on their visual features, the next, and last, step considers the temporal information. A temporal-aware reranking introduces the concept that the lost object may not be in the location with the best visual match with the query, but in the last location where it was seen.

Two rankings \( R_C \) and \( R_D \) are built by reranking the elements in \( C \) and \( D \), respectively, based on their time stamps. The final ranking \( R_t \) is built as the concatenation of \( R_t = [R_C, R_D] \) (which considers the best candidate to be at the beginning of the sequence). Thus, \( R_t \) always contains all the images in \( I \) and we ensure that any relevant image will appear somewhere in the ranking, even after the thresholded cases. We propose two strategies to exploit the time stamps of the images:

a) Decreasing Time-Stamp Sorting: This is the most simple approach we can consider at this point. Just a simple reordering of the \( C \) and \( D \) sets to build the \( R_C \) and \( R_D \) rankings from the latest to the earliest time-stamp. This configuration will be applied in all experiments, unless otherwise stated.

b) Interleaving: This other approach introduces the concept of diversity. We realized that the rankings tend to present consecutive images of the same moment when using the straightforward sorting. This is expected behavior due to the high visual redundancy of neighboring images in an egocentric sequence. As the final goal of

\(^1\) cosine similarity(a,b) = \cos(a,b) Note that it is always between 0 and 1 as vectors have non-negative components.
this work is determining the location of the object, showing similar and consecutive images to the user is uninformative. By introducing a diversity step, we force the system to generate a rank list of diverse images, which may increase the chances of determining the object location by looking at the minimum elements in the ranked list.

Our diversity-based technique has its basis in the interleaving of samples. In digital communication, interleaving is the reordering of data that is to be transmitted so that consecutive samples are distributed over a larger sequence of data in order to reduce the effect of burst errors. Adapting it to our domain, we interleave images from different scenes to put a representative of each scene early in the ranking. Thus, if the first candidate is not relevant, we avoid the second to be from the same scene and, therefore, it is more likely to be relevant. Figure 4 depicts the temporal diversity strategy. The algorithm proceeds as follows:

1. Make a list with all the images in $I$ sorting by their timestamp in decreasing order. That is, the later image the first. For each image $i \in I$ it must be known whether it belongs to $C$ or $D$. Such as,

$$O = \{C_{t_{n-1}}^{C}, ..., C_{t_{m-1}}^{C}, C_{t_{l}}^{D}, ..., C_{t_{l-1}}^{D}, ..., C_{t_{k}}^{C}, C_{t_{k-1}}^{D}, ..., i_l \}$$

2. Split into sub-lists using the transitions $C \rightarrow D$ or $D \rightarrow C$ as a boundary.

3. Build a new list $R_C$ by adding the first image of each sub-list containing elements in $C$ maintaining timestamp in decreasing order. Then, the second image of each sub-list and so on. Thus, $R_C = \{C_{t_{n-1}}^{C}, C_{t_{m-1}}^{C}, C_{t_{l}}^{D}, C_{t_{l-1}}^{D}, ..., C_{t_{k}}^{C}, C_{t_{k-1}}^{D}, ..., i_l \}$. Build $R_D$ analogously.

4. Concatenate $R_C$ and $R_D$ to obtain the final ranking $R_t = [R_C, R_D]$.

4. EXPERIMENTS

The proposed system was trained and evaluated in a subset of images from the NTCIR-Lifelog dataset [13] according to the Mean Reciprocal Rank (MRR) [6]. The details and results are presented and discussed in this section.

4.1 Datasets

Our experiments used the NII Testbeds and Community for Information access Research (NTCIR) Lifelog dataset, which is composed of a total of 88,185 images acquired by 3 people using an Autographer camera during 90 days, 30 days per person. In our experiments, though, we only considered one of the users. The Autographer camera used in the NTCIR-Lifelog dataset uses a wide angle lens, a feature which resulted helpful as the images where more likely to include personal objects.

4.1.1 Definition of Queries

When performing a search the system needs an input of some images of the object in order to look for it. To carry out the experiments, and after doing an exhaustive analysis of the dataset, we decided to work with four object categories: mobile phone, laptop, watch, and headphones. The set $Q$ was built containing five images of the own dataset for each category. The whole object was present in these images and occupied most of it.

The query images were manually annotated with a bounding box to assess the Hard Bounding Box (HBB) and Soft Bounding Box (SBB) configurations. In our dataset, this corresponded to a total of 25 bounding boxes. We consider that this scenario is realistic because, if necessary, a user could be ask to manually annotate five images of the object he is looking for. However, the results presented later in Section 4.4 show that this task may not be even necessary.

4.1.2 Annotation of the Dataset

We decided to build the annotations of the dataset following this guideline: “Consider as relevant those images that would help to find out where was the last time that the camera saw the object”. This strategy made us considering as relevant all the images that depicted both the location and the object. Any of them would help the user to find his object.

4.2 Evaluation metric

To assess the performance of the system, an evaluation metric must be chosen to be able to quantitatively compare how different configurations perform. This metric should be as realistic as possible and must have the ability to measure exactly whether or not the system helps the user when he or she looks for the objects.

The Mean Reciprocal Rank (MRR) [6] is the average of the reciprocal ranks of results for a sample of queries $Q$, being the reciprocal rank of a query response the inverse in the rank of the first relevant answer $q^*$. For given a day $d$, its mathematical expression is:

$$\text{MRR}_d = \frac{1}{|Q_d|} \sum_{q \in Q_d} \frac{1}{q^*}$$

Mean Reciprocal Rank is associated with a user model where the user only wishes to see one relevant document. We have defined the Averaged-MRR (AMRR) to refer to the average of MRRs obtained across test all days. Given a set of days $D = \{d_1, d_2, ..., d_k\}$ the expression of the Averaged Mean Reciprocal Rank is

$$\text{AMRR} = \frac{1}{|D|} \sum_{d \in D} \text{MRR}_d = \frac{1}{|D|} \sum_{d \in D} \frac{1}{|Q_d|} \sum_{q \in Q_d} \frac{1}{q^*}$$

4.3 Training

The 30 days of data available for one user were used in the following way: the queries were defined using 3 days, the training partition included 9 different days, and the testing was performed on the remaining 15 days. The remaining 3 days were discarded as they did not include enough quality appearances of the objects to be considered.

The values that we wanted to train were the construction of the codebook for the visual words as well as the thresholds used in both techniques described in 3.3, TVSS and NNDR.

- **Visual Words Codebook**: The BoW framework defined by [23] requires building a visual codebook in order to map vectors to their nearest centroid. This codebook was built using...
Figure 5: Training the thresholds using saliency maps for \( g \).

Figure 6: Set of images \( Q \) used to build the query vector \( \vec{q} \) for the category mobile phone.

\( k \)-means clustering. We used an accelerated algorithm based on approximate nearest neighbors on local CNN features to fit a codebook with 25,000 centroids as in [23].

- **Threshold values** \( \nu_{th} \) and \( \rho_{th} \): To predict what would be the best value for these parameters, the same procedure was applied for both. We performed a sweep from 0 to 1 with a step-size of 0.01. For each of these values the MRR was computed and averaged across the 9 days that composed the training set. Therefore, this AMRR can be understood as a function of the threshold, so an optimal argument can be chosen. Figure 5 shows the curves obtained and the optimal values chosen when training with saliency maps for the \( g \) function. When training using other configurations for the \( g \) function, the optimal thresholds \( \rho_{th} \) and \( \nu_{th} \) remained in the same values despite AMRR being slightly different.

### 4.4 Testing

The different methods presented in Section 3 are assessed in this section over a test set of 15 days from the NTCIR Lifelog dataset. Table 1 summarizes the configuration options for the different stages of the pipeline. The study aims at identifying which is the impact of each of the proposed methods in the complete solution.

Tables 2, 3 and 4 contain the AMRR values obtained for all possible configurations, being *Time Sorting* the baseline. Visual Ranking is included to understand the impact of the proposed temporal reranking stages with respect to the visual search obtained with [23].

Comparing the results obtained with any of the system configurations (the four last columns) versus the intermediate stage (the visual ranking \( R_V \)) and the defined baseline (the temporal sorting), we can conclude that all proposed methods both the visual search and the diversity-based reranking improve the AMRR. In other words, all configurations provide a faster option to find the last appearance of the lost object than brute-force approach of just browsing backwards in time through the sequence of egocentric images.

Comparing each approach with or without temporal diversity indicates a gain in all cases but in the SBB-NNDR one of Table 2, where performance slightly drops. Our study reveals that, in general, a temporal interleaving of the results is a good choice. This is true in all the best configurations of Tables 2, 3 and 4.

\(^2\)I stands for Interleaving

<table>
<thead>
<tr>
<th>Method Options</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Full Image (FI)</td>
</tr>
<tr>
<td>Target database</td>
<td>Center Bias (CB)</td>
</tr>
<tr>
<td>Visual Ranking</td>
<td>Saliency Map (SM)</td>
</tr>
<tr>
<td>Thresholding criterion</td>
<td>Nearest Neighbors (NNDR)</td>
</tr>
<tr>
<td>Ranking</td>
<td>Time Sorting</td>
</tr>
<tr>
<td></td>
<td>Interleaving (I)</td>
</tr>
</tbody>
</table>

Table 1: Configuration parameters summary.
A comparison between Table 2 and 3 shows that weighting the convolutional features of the target dataset with a central bias does not improve the results obtained when using the full image query, despite this strategy has been used in other works related to salient object detection [20]. We suspect that this is due objects in egocentric images not being located in the center of the image as often as they are for intentionally taken photographs. This same conclusion was reached in [3].

Focusing in Table 4 indicates that weighting the convolutional features of the target images with a saliency map is only beneficial when the full query image is considered, actually providing the best results among all configurations. However, saliency maps decrease the performance when a hard bounding box defines the query, and does not introduce much changes when a soft bounding box is considered. In other words, focusing on the local information of the object in the query image may not be beneficial, and exploiting its context may help. Notice though that using the full query image is only the best of the three query configurations when the convolutional features of the target images are weighted by the saliency maps. We hypothesize that the saliency maps, apart from identifying the local features of the object in the target image, it also emphasizes other features in the background that boost the matching with the full query images. In these later cases, we would be exploiting the case when an object tends to appear in the same locations, somehow providing a prior for our system. When an object is lost, looking at the places where we have used it frequently in the past may be beneficial, and the features characterizing the location are found outside the local query mask.

5. CONCLUSIONS

The main objective of this work was to design a retrieval system to find personal objects in egocentric images. Compared to the proposed baseline, the contributions reported in this document have shown that the system is helpful for the task. We believe these results might be useful as a baseline for further research on this field.

An interesting observation of this work is the fact that, despite being a common strategy in many other tasks, applying a central bias weighting did not improve results, but weighting with saliency maps improved performance significantly.

As a future work, we suggest to explore different approaches for the temporal reordering stage that might improve the system performance. Referring to the visual part, fine-tuning could be performed to adapt the network to the egocentric images and improve its improve the accuracy when extracting the local convolutional features.

6. ACKNOWLEDGMENTS

The main author developed this work thanks to the financial support of the Erasmus+ Programme for Student Mobility in the European Union, Generalitat de Catalunya and Centre de Formació Interdisciplinaria Superior (CFIS). This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant number SFI/12/RC/2289. The Image Processing Group at the UPC is a SGR14 Consolidated Research Group recognized and sponsored by the Catalan Government (Generalitat de Catalunya) through its AGAUR office. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeForce GTX Titan Z used in this work.

7. REFERENCES

[4] Jaime Carbonell and Jade Goldstein. The use of mmm, diversity-based reranking for reordering documents and


