ABSTRACT

Recent progress in the field of content-based image retrieval has enabled camera-based indoor positioning. The matching of smartphone recordings with a database of geo-referenced images allows for meter accurate infrastructure-free localization. In mobile scenarios, however, three major constraints have to be considered: limited computational resources of mobile devices, limited network capacity and the need for scalability in large buildings. To address these issues, we modify the state-of-the-art Vector of Locally Aggregated Descriptors (VLAD) image signature to work with recently emerging binary feature descriptors. We show that this results in a substantial reduction in the overall computational complexity, which enables the matching of image signatures directly on the mobile device. The specific properties of this signature form the basis of our proposed scalable streaming approach that preemptively loads image signatures of reference images in the vicinity of the user onto the mobile device to mitigate the effect of network latency. In order to provide efficient streaming, we compress the signatures by exploiting the similarities of spatially neighboring reference images. In combination, the contributions of this paper lead to an indoor localization system, which allows instantaneous camera-based indoor positioning with very low requirements on the available network connection.

Index Terms— Binary Features, Vector of Locally Aggregated Descriptors, Visual Localization, Mobile Visual Search

1. INTRODUCTION

Camera-based localization is a promising approach for accurate indoor positioning without the need for expensive infrastructure like radio beacons. Images taken by a mobile device are matched to previously acquired reference images with known position and orientation. By determining the visually most similar reference image, the position of the device is retrieved. With vast improvements in the field of content-based image retrieval (CBIR), state-of-the-art algorithms already provide the tools for an accurate localization framework. Some of these algorithms are, however, quite demanding in terms of their computational complexity and therefore not suited to run on mobile devices, where computational power and battery life are limiting factors and where network connectivity is still an important issue. A possible approach to tackle the latter problem would be to store the whole reference database on the mobile device. While this approach requires no network connection at all, it would hardly make sense to preemptively download the reference databases of all buildings of a city and moreover would exceed the storage capacity of today’s mobile devices. To this end, client server architectures have been proposed where the mobile device sends compressed query information to a server, which returns the location of the device. This can be done by either sending the JPEG compressed query image [1], by compressing the extracted query features [2], or by quantizing the query features into a bag of visual words and sending only the respective indices [3]. All of the aforementioned approaches trade off data traffic against required computational power at the mobile device [4]. The downsides of these approaches are the permanently required network connectivity, the unfavorable impact of network delay on the localization speed and the inefficient use of the network resources, as mobile networks usually provide more data rate on the downlink than on the uplink. An approach to overcome these limitations is to preload the currently required data to the mobile device, for example using so-called partial visual vocabularies [5, 6]. This approach still consumes a significant amount of computational power and transmission rate for providing seamless and close-to-realtime localization services in extensive indoor environments. In order to jointly address the problem of computational burden and memory footprint, we modify VLAD [7, 8] to work with low-complexity binary features such as BRIEF, ORB or BRISK [9, 10, 11]. The proposed image signature enables a very compact representation, which is the basis for our main contribution, an efficient streaming and compression algorithm for image signatures. To become independent of network latency, we preemptively load the image signatures of reference images in the vicinity of the user onto the mobile device, which are subsequently available for offline localization. This streaming approach allows for a new compression scheme that exploits the visual similarities of spatially neighboring reference images to further reduce the required transmission rate.

The remainder of this paper is organized as follows. In Section 2, we discuss related work on image signatures in content-based image retrieval. In Section 3, we introduce our image signature based on binary feature descriptors, in the following coined BVLAD. Section 4 describes the streaming architecture and the novel compression algorithm. An evaluation of BVLAD and the proposed streaming approach is presented in Section 5. Finally, in Section 6 we conclude the paper and provide an outlook on possible extensions.

2. RELATED WORK

The abstraction of an image by a histogram of local image features forms the basis of state-of-the-art content-based image retrieval engines. Mapping the set of local image features to a fixed-length vector using Fisher Kernels was first proposed by Perronnin et al. in [12, 13]. Their approach is based on the assumption that the generative distribution model of local features within images can be approximated by a Gaussian mixture model. Applying the Fisher Kernel results in similar performance as the state-of-the-art Bag of Words (BoW) approach originally proposed by Sivic et al. [14]. Later, the Vector of Locally Aggregated Descriptors was introduced by Jégou et al. [7], which resembles a simplified version of the Fisher Kernel. They propose to learn a small visual vocabulary consisting of visual words, i.e., a set of representative feature descriptors
for a specific database. Subsequently, they calculate the residual between each descriptor and its nearest visual word. After aggregation, the residuals of all visual words are concatenated to form a fixed-length image representation yielding comparable results to BoW-based approaches while using an order of magnitude less memory. Recently, several enhancements to the basic VLAD have been proposed. Chen et al. propose in [8] to strongly quantize the VLAD representation to binary values aiming at rapid image retrieval on mobile devices. A more recent paper from Jégou et al. [15] provides a closer insight into the relation of Fisher Kernels and Bag of Words. They additionally suggest more types of aggregation methods, applying an initial PCA to the feature vectors and employing a product quantizer [16] to match individual parts of the VLAD against the database. Arandjelović et al. [17] propose further approaches for residual normalization, vocabulary adaptation for different datasets and multiple VLAD representations covering subregions of the image. The extensive evaluation of VLAD by Delhumeau et al. in [18] focuses on the residual normalization and proposes to apply the PCA independently on parts of the residual vector, which is referred to as local coordinate system.

3. RESIDUAL VECTOR USING BINARY FEATURES

In previous work, VLAD and Fisher Kernels have only been applied in combination with features like SURF [19], SIFT [20] or CHoG [2], which rely on floating point representations. To speed-up the overall image retrieval process, we adapt the basic ideas of VLAD to form an image signature that performs best in conjunction with binary features like BRIEF, ORB or BRISK.

3.1. Residual Calculation

The first step towards the proposed BVLAD representation is learning a small visual vocabulary \( C = \{ c_1 \ldots c_k \} \) using the k-means algorithm. Experiments show comparable results for binarized versions of the clusters or a fast binary k-majority algorithm [21]. This can be explained through the coarse quantization where the absolute values have minor influence compared to the overall distribution. After assigning every local feature descriptor \( \mathbf{x} \) to its closest visual word \( c_i = \text{NN}(\mathbf{x}) \), the residual vector \( \mathbf{r}_i \) for visual word \( c_i \) can be calculated by accumulating the difference between the descriptor and the centroid:

\[
r_i = \sum_{\mathbf{x} : \text{NN}(\mathbf{x}) = c_i} (c_i - \mathbf{x})^T, \quad \forall i \in 1 \ldots k
\]

Next, a power-law (eq. 2) is applied to the residual \( \mathbf{r}_i \) to cope with bursty components. In contrast to non-binary features such as SURF, binary features do not exhibit peaky components. Nevertheless, the power law reduces the influence of frequently re-occurring feature descriptors stemming from similar shapes like doors and windows:

\[
r'_{i,j} = r_{i,j}^\alpha, \quad \forall i \in 1 \ldots k, \quad \forall j \in 1 \ldots l
\]

Where \( r_{i,j} \) denotes the j-th entry of the residual vector associated with visual word \( c_i \) and \( l \) is the length of the binary descriptor. In our setup, we employ a \( l = 256 \) dimensional BRIEF descriptor. In the experimental evaluation, the best performance was achieved using a value of \( \alpha = 0.8 \). Additionally, the difference vectors are normalized using the L2-norm to balance the contribution of each cluster to the final vector, which is called intra-normalization [17].

\[
r''_i = r'_i / \| r'_i \|, \quad \forall i \in 1 \ldots k
\]

To form a residual vector, the accumulated differences are concatenated as follows:

\[
r = [r''_1 \ldots r''_k]
\]

With a binary feature descriptor length \( l \), the dimensionality \( n \) of the row vector \( \mathbf{r} \) becomes \( n = k \cdot l \). Finally, the vector \( \mathbf{r} \) is normalized to unit length. Assuming that every part \( r''_i \) has descriptors associated (no zero values) this corresponds to a division by \( \sqrt{k} \):

\[
r = \frac{r}{\| r \|} = \frac{r}{\sqrt{k}}
\]

3.2. Dimensionality Reduction

The resulting \( n \) dimensional vector \( \mathbf{r} \) can be further reduced in order to minimize the memory footprint of the final image signatures. We follow the concept of the local coordinate system [18], which separately applies a PCA on all subsections \( \mathbf{r} \), associated with the visual word \( c_i \) of the residual vector. This alleviates the influence of the correlation among the binary tests in the BRIEF pattern, as according to [10], the first components of the PCA includes the essential information about the BRIEF feature descriptor. Hence, only the \( l' \) principal components with highest eigenvalues within each subsection are used to transform the residual vector, resulting in a \( n' = k \cdot l' \) dimensional image signature \( \mathbf{r}' \). Finally, the zero-centered vector is binarized to the final vector \( \mathbf{b} \) by thresholding. This results in the final binary image signature, which we refer to as BVLAD.

3.3. Matching

The main advantage of the binarization, apart from a reduced memory footprint, is a very fast matching process using the Hamming distance \( h(\mathbf{x}, \mathbf{y}) \). This distance can be computed very efficiently using intrinsic processor instructions, which are also available on modern mobile devices. For the ranking process, we match every subsection \( \mathbf{b} \) of the image signature separately. The final score \( f \) for query image \( q \) and database image \( d \) is calculated as:

\[
f(\mathbf{b}_q, \mathbf{b}_d) = \frac{1}{\sqrt{|F_q||F_d|}} \sum_{i \in F_q \cap F_d} 1 - \frac{h(\mathbf{b}_{q,i}, \mathbf{b}_{d,i})}{l'}
\]

Where \( F_q \) and \( F_d \) are the sets of visual words observed in the database and query image, respectively. When dealing with larger datasets, e.g., whole cities, we employ a locality sensitive hashing algorithm, which provides fast approximate nearest neighbor search.

4. MOBILE LOCALIZATION

4.1. Reference Database

In our mobile localization scenario, the reference database contains images captured inside buildings and can be queried with an image taken by a mobile device, using content-based image retrieval. The position information attached to the most similar reference view serves as the location estimate. In order to build a meter-accurate localization system with the possibility to also determine viewing angles, we employ the virtual view approach proposed by Huitt et al. [22]. The virtual views are artificially generated sets of reference views at a grid with 1 meter distance and annotated with position and orientation information. The views used in our experiments are based on a recent version of the TUM Indoor Dataset [23] and cover 3, 431 positions. At every position 16 different viewing angles are available, resulting in a total of 54, 896 reference views.
4.2. Streaming Approach

Instead of performing the matching at the server, we preemptively load the BVLAD representation of the reference images in the near vicinity onto the mobile device. This allows us to become independent of network latency and take advantage of the typically faster downlink. Prior knowledge about the coarse location of the mobile device is usually available via Wi-Fi, Cell-ID based localization or via the last GPS fix [6]. Based on the assumption that spatially neighboring, and hence partially overlapping reference images result in similar image signatures, we further reduce the memory footprint by exploiting this redundancy. To this end, we subdivide the database into tiles of size $m \times m$ meters. The tiles are further subdivided according to the viewing angle such that every tile holds $v$ out of 16 adjacent viewing angles. This ensures that the resulting group of reference images shares overlapping image regions, which can be exploited by the compression approaches described in the following.

4.3. Lossless Compression

The compression of binary features has been investigated by Ascenso et al. [24]. They employ an inter-coding scheme, which sorts a set of binary features according to their Hamming distance starting with the feature exhibiting the highest detector response. They apply bitwise XOR operations between a feature vector and its successor to exploit the similarities followed by arithmetic coding. According to their evaluation, this results in rate savings of up to 32%. Redondi et al. [25] propose a lossless intra-coding using conditional entropy to represent a 128-dimensional binary feature descriptor with 80 bits. In contrast to these approaches, we base our compression approach on the visual similarity of spatially neighboring reference images. Considering the design of the proposed BVLAD as described before, it can be assumed that signatures of grouped images contain similar subsections (parts of BVLAD that are associated with the visual word $v_i$). The efficient coding of recurring subsections in a group of signatures can be performed via the deflate algorithm, which is a combination of the LZ77 algorithm (duplicate string elimination) and Huffman coding. By replacing (partially) recurring strings among the grouped image signatures the aforementioned compression schemes are outperformed by our approach.

4.4. Lossy Compression

While maximum retrieval performance and localization accuracy is the main goal, it can be questioned whether lossless compression is actually required. Hence, to further reduce the data rate, we propose a lossy compression method using visual prototypes for each group of signatures. The approach can be considered as a location adaptive version of the product quantizer with asymmetric distance calculation [16]. For each individual subsection $s_i$ associated with the visual word $v_i$, we train a binary version of a $k$-majority algorithm [21] using the signatures contained in one group. The result is a set of $p$ prototypes for each of the $k$ subsections. As the prototypes of each subsection can be independently combined with each other, we obtain $p^k$ possibilities to assemble different image signatures. Hence, we can approximate the original image signature $s$ in the group by using the prototype $p$ for each subsection $i$ that minimizes the Hamming distance. Eventually, for each group, only the used location adapted prototypes have to be transmitted encompassed by the respective indices $a_i$ to approximate the original signatures at the mobile device, as shown in Figure 2. The intuition behind this approach is similar to the deflate algorithm, which eliminates (partially) recurring subsections among a group of BVLAD vectors. This lossy approach, however, ignores minor differences among similar subsections by quantizing the subsections to one of the $p$ prototypes.

5. RESULTS

5.1. Query Dataset

We recorded a query set of 128 images captured by a Samsung Galaxy S3 smartphone with manually annotated position information. For a detailed evaluation, the query set has been split into classes: high texture, low texture, hallways, ambiguous objects and building structure, where each query can be assigned to more than one class (Fig. 1a-1e). To make our evaluation comparable to previous work, we ignore the orientation information. Therefore a reference image is considered to be a match, if it is located within a maximum distance of 5 meters to the ground truth location of a query.
In order to identify optimal parameters for the approach described above, several experiments are conducted with varying settings. In Table 1, we present the impact of the vocabulary size \( k \) by evaluating the precision at first, third and fifth rank searching the whole database. The precision denotes the fraction of relevant results from all retrieved results up to a rank \( r \). A precision of 1.0 at rank \( r \) is achieved, if all top \( r \) results are a match. For this experiment, the dimensionality reduction is fixed to \( l' = 64 \). The best performance can be achieved at large vocabulary sizes with the downsides of increasing matching time, memory consumption and required data rate. For the following experiments, the vocabulary size is fixed to \( k = 256 \) as a trade-off between retrieval precision and speed. Table 2 illustrates the impact of the dimensionality reduction using the \( l' \) largest eigenvalues. With a small loss of performance, the signature size can be reduced by a factor of 4 resulting in 2048 bytes per signature. In Table 3, we provide a comparison between BoW using histograms of visual words and the BVLAD representation. The most challenging class is low texture, as there are hardly enough reliable features. Also extensive hallways and repetitive objects throughout the building result in lower precision. In comparison, BVLAD outperforms the BoW in high-textured scenarios whereas it is inferior in low-textured environments. In order to measure the retrieval time, we use an Intel Core i7 x64 system with 3.4 Ghz and obtain the timings given in Table 4. To speed up the retrieval time, we replace the linear search (LS) with a locality-sensitive hashing (LSH) algorithm using conservative settings [26] yielding similar results at a speed-up by a factor of 2 and outperforming BoW by a factor of 4. In Table 5, we present results for considering only the reference images within a radius \( d \) meters to the query location exploiting prior knowledge.

Table 2. Impact of the local coordinate PCA for \( k = 256 \).

<table>
<thead>
<tr>
<th>Classes</th>
<th>Bag of Words</th>
<th>BVLAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Texture</td>
<td>0.65 / 0.63 / 0.62</td>
<td>0.74 / 0.72 / 0.69</td>
</tr>
<tr>
<td>Low Texture</td>
<td>0.52 / 0.49 / 0.48</td>
<td>0.49 / 0.44 / 0.42</td>
</tr>
<tr>
<td>Hallways</td>
<td>0.59 / 0.58 / 0.54</td>
<td>0.48 / 0.43 / 0.41</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>0.59 / 0.52 / 0.54</td>
<td>0.48 / 0.51 / 0.50</td>
</tr>
<tr>
<td>Building Structure</td>
<td>0.61 / 0.54 / 0.54</td>
<td>0.53 / 0.52 / 0.49</td>
</tr>
<tr>
<td>Total</td>
<td>0.59 / 0.56 / 0.55</td>
<td>0.62 / 0.58 / 0.55</td>
</tr>
</tbody>
</table>

Table 3. Precision per class at first, third and fifth rank. BVLAD with \( k = 256, l' = 64 \), Bag of Words with vocabulary size 200000.

5.2. Location Retrieval

Table 4. Average timing of the processing steps per query with 1300 features. SURF timings with equal feature numbers are included.

<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAST Detector</td>
<td>0.88 ms</td>
</tr>
<tr>
<td>BRIEF Descriptor</td>
<td>4.77 ms</td>
</tr>
<tr>
<td>SURF Detector</td>
<td>232.6 ms</td>
</tr>
<tr>
<td>SURF Descriptor</td>
<td>248.0 ms</td>
</tr>
<tr>
<td>BVLAD Calculation</td>
<td>14.12 ms</td>
</tr>
<tr>
<td>BVLAD Matching (LS)</td>
<td>100.17 ms</td>
</tr>
<tr>
<td>BVLAD Matching (LSH)</td>
<td>53.74 ms</td>
</tr>
<tr>
<td>BoW Matching (LSH)</td>
<td>220.93 ms</td>
</tr>
</tbody>
</table>

Table 5. Results including only images within a radius \( d \).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>( k )</th>
<th>( l' )</th>
<th>Gain</th>
<th>( P1 )</th>
<th>( P3 )</th>
<th>( P5 )</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossless</td>
<td>64</td>
<td>64</td>
<td>27%</td>
<td>0.49</td>
<td>0.49</td>
<td>0.46</td>
<td>0</td>
</tr>
<tr>
<td>Lossy</td>
<td>64</td>
<td>64</td>
<td>67%</td>
<td>0.49</td>
<td>0.46</td>
<td>0.43</td>
<td>811</td>
</tr>
<tr>
<td>Lossy 5m</td>
<td>64</td>
<td>64</td>
<td>80%</td>
<td>0.50</td>
<td>0.48</td>
<td>0.46</td>
<td>1167</td>
</tr>
<tr>
<td>Lossless</td>
<td>256</td>
<td>64</td>
<td>52%</td>
<td>0.62</td>
<td>0.58</td>
<td>0.55</td>
<td>0</td>
</tr>
<tr>
<td>Lossy 5m</td>
<td>256</td>
<td>64</td>
<td>69%</td>
<td>0.63</td>
<td>0.58</td>
<td>0.55</td>
<td>1622</td>
</tr>
<tr>
<td>Lossy 5m</td>
<td>256</td>
<td>64</td>
<td>84%</td>
<td>0.56</td>
<td>0.52</td>
<td>0.51</td>
<td>4342</td>
</tr>
</tbody>
</table>

Table 6. Different compression schemes with 4 prototypes, 2 adjacent views and area of 3 x 3 meters per group except Lossy 5m. Error denotes the average number of false bits per reconstructed signature.

5.3. Compression

The best performance of the proposed compression scheme is achieved when limiting the tile size to 3 x 3 meters and 2 neighboring viewing angles, as shown in Table 6. Increasing the tile size or adding more viewing angles increases the diversity of the visual appearance and more prototypes are required for an adequate reconstruction. For lossless compression, we achieve a compression gain of 52 % for \( k = 256, l' = 64 \). For lossy compression, however, we achieve a compression gain of 69 % with no significant impact on the retrieval results. Moreover, for \( k = 64, l' = 64 \), a compression gain of 80 % can be achieved with only slight impact on the results. With increasing \( k \), more visual words have no feature attached and therefore share the same binary pattern (e.g. zeros), which improves the compression ratio. The results show the potential of our appearance-driven compression, which allows a significant data rate reduction with no degradation in retrieval performance.

6. CONCLUSION

We present a scalable mobile camera-based localization system. To this end, we propose a modified version of Vector of Locally Aggregated Descriptors, which is based on binary features and jointly addresses the problem of limited computational capacity, as well as the required memory footprint. For rapid matching, we binarize the resulting vectors, which leads to the BVLAD representation. Our main contribution is an efficient streaming approach, which employs prior knowledge to stream only relevant image signatures within a certain radius. Our approach includes a lossy compression scheme that exploits visual similarities between spatially related views and generates location adapted prototypes for every subsection of the image signature. This results in a significant reduction in data rate down to 20 % of the uncompressed size, while maintaining the full retrieval performance. The proposed system may be combined with any of the recently emerging binary feature descriptors.

7. ACKNOWLEDGEMENT

This research project has been supported by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the reference 50NA1307. The authors also acknowledge the support from a Qualcomm Innovation Fellowship.
8. REFERENCES


