Absolute Spatial Context-aware Visual Feature Descriptors for Outdoor Handheld Camera Localization

Overcoming Visual Repetitiveness in Urban Environments

Daniel Kurz[1], Peter Georg Meier[1], Alexander Plopski[2] and Gudrun Klinker[3]
[1]*metaio GmbH, Munich, Germany*
[2]*Osaka University, Osaka, Japan*
[3]*Technische Universität München, Munich, Germany*

Keywords: Visual Feature Descriptors, Repetitiveness, Camera Localization, Inertial Sensors, Magnetometer, GPS.

Abstract: We present a framework that enables 6DoF camera localization in outdoor environments by providing visual feature descriptors with an Absolute Spatial Context (ASPAC). These descriptors combine visual information from the image patch around a feature with spatial information, based on a model of the environment and the readings of sensors attached to the camera, such as GPS, accelerometers, and a digital compass. The result is a more distinct description of features in the camera image, which correspond to 3D points in the environment. This is particularly helpful in urban environments containing large amounts of repetitive visual features.
Additionally, we describe the first comprehensive test database for outdoor handheld camera localization comprising of over 45,000 real camera images of an urban environment, captured under natural camera motions and different illumination settings. For all these images, the dataset not only contains readings of the sensors attached to the camera, but also ground truth information on the full 6DoF camera pose, and the geometry and texture of the environment. Based on this dataset, which we have made available to the public, we show that using our proposed framework provides both faster matching and better localization results compared to state-of-the-art methods.

1 INTRODUCTION

Video-see-through Augmented Reality (AR), as the concept of seamlessly integrating virtual 3D content spatially registered into imagery of the real world in real time, is currently becoming ubiquitous. It recently made its way from research labs to the mass market. A fundamental enabler for AR moving mainstream is affordable off-the-shelf hardware such as camera-equipped mobile phones and tablet PCs. The dense integration of a computer with a display, cameras, different communication interfaces, and a variety of sensors make these devices interesting for AR.

One of the most important challenges towards the everyday usage of handheld AR is precise and robust camera localization outdoors. Pose estimation, which is based only on information from sensors such as GPS, compass and inertial sensors, is currently being used in AR browsers. The precision of this is controlled by environmental conditions and is usually not enough for pixel-precise registration of overlays in the camera image, as shown in figure 1 (right).

Visual localization and tracking is very well suited to provide very accurate registration, and is frequently used for camera tracking in desktop-sized environments. In particular, simultaneous tracking and mapping (SLAM) systems are commonly used, e.g. (Klein and Murray, 2009), to reconstruct a sparse 3D map of the environment whilst tracking. The use case targeted in this paper, however, requires an offline learned model of the environment because we need to work in an absolute and known coordinate system. This enables overlaying landmarks or signs for pedestrian navigation correctly registered with the camera image, whereas SLAM operates in an arbitrary coordinate system. The difference between SLAM and our method is not only that we use an offline learned map, but also that this map might be many weeks old and the visual appearance of the environment might have changed since then.

While SLAM applications usually assume a static environment, there are different challenges to tackle when going outdoors. Illumination and weather may be subject to change and parts of the environ-

Kurz D., Meier P., Plopski A. and Klinker G..
Absolute Spatial Context-aware Visual Feature Descriptors for Outdoor Handheld Camera Localization - Overcoming Visual Repetitiveness in Urban Environments.
DOI: 10.5220/0004683300560067
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 56-67
ISBN: 978-989-758-004-8

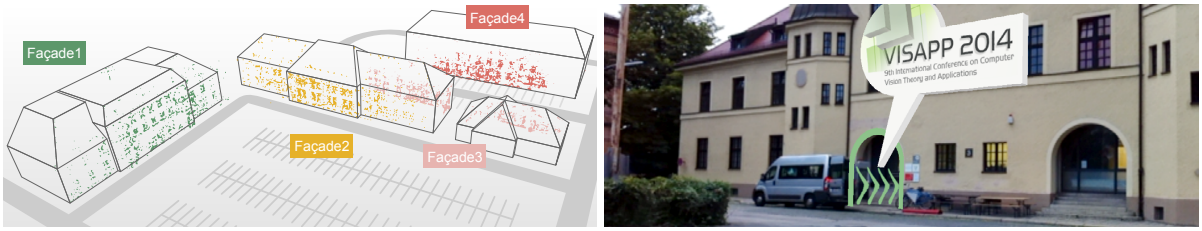

Figure 1: Our framework localizes a camera with respect to sparse feature map and exploits a coarse environment model together with sensor readings to aid feature detection, description and matching (left). The obtained pose can finally be used in outdoor AR applications (right).

ment, such as parked cars and pedestrians, frequently change and might occlude static parts of the environment. Another very important challenge is handling repetitive visual structures which are ubiquitous in urban and man-made environments. This was for example discovered in (Arth et al., 2012).

The dataset presented in this paper covers all the above aspects, for it comprises of real camera images of a real urban environment taken at different points in time. Additionally, the proposed methodology for creating this ground truth evaluation dataset allows for easy expansion in the future.

Our main contribution in this paper is to tackle the problem of repetitive visual features by making the description of these features aware of their Absolute Spatial Context (ASPAC), i.e. position, orientation and scale in the world coordinate system. Thereby, we enable discrimination between visually indistinguishable features, which is crucial for visual outdoor camera localization.

2 RELATED WORK

2.1 Outdoor Camera Localization

A common approach to visual localization is to use local image feature descriptors, such as SIFT (Lowe, 2004), which describes features in a way that is invariant to changes in (in-plane) rotation and scale. These descriptors can be used for place recognition, e.g. (Knopp et al., 2010), to determine the coarse position of what is shown in a query image by finding a corresponding database image with a known position. Other approaches use SIFT features to match against reference descriptors with known 3D positions associated to them, e.g. as a result of structure from motion (SfM) methods on large sets of images (Irschara et al., 2009) or video sequences taken with an omnidirectional camera (Ventura and Höllerer, 2012).

One of the most important challenges when using such local image features in urban outdoor environments is repetitive features, as also mentioned in (Knopp et al., 2010), (Ventura and Höllerer, 2012), and (Arth et al., 2012). Certain objects, such as trees or traffic signs, occur at many different locations, and therefore may lead to confusions in visual place recognition. (Knopp et al., 2010) propose to identify such confusing features in their database and suppress them, which will significantly improve recognition performance.

When the dataset mainly consists of building façades, there are usually many similar looking features spatially close to each other, such as the windows. (Baatz et al., 2012) visually estimate vanishing points in query images under the assumption that vertical and horizontal edges are predominant. After rectification of a façade, they then perform pose estimation separately for the two axes on the façade. As a result, features that were matched in the correct story, but with a wrong window in that row, can still contribute to the correct pose. Additionally, they use upright feature descriptors that increase their distinctiveness at the expense of invariance to rotation, which is not needed on rectified images. (Kurz and Benhimane, 2011) proposed to use the measured direction of gravity for descriptor orientation when inertial sensors are attached to the camera, which has a similar effect without the need for expensive vanishing point estimation.

Most current mobile phones are additionally equipped with a GPS receiver and a digital compass, which was exploited by (Arth et al., 2012) to partition 3D reference features according to their position and orientation (camera heading). They then match camera features only against those reference features located in the cell where the camera is according to GPS. They also only match against reference features resulting from camera views with a heading similar to the one currently measured with the attached compass. This increases both the robustness and speed of their 6DoF localization method.

Not only the position and orientation of visual features can add distinctiveness to their representation, but also their physical scale can as well. (Smith et al.,

2012) make use of combined range-intensity data, allowing the extraction and description of features at a physical scale. Thereby, confusions between similar looking features at different physical scales can be avoided. (Fritz et al., 2010) use EXIF information stored with digital images to gain information on the metric size of objects shown in the image.

Another approach to wide-area localization proposed by (Reitmayr and Drummond, 2007) uses the coarse pose obtained from GPS, compass and inertial sensors as a prior for model-based tracking of an urban environment. However, as their tracker requires the prior to be much more accurate than the precision usually obtained from GPS, they attempt initialization with a set of prior poses sampled around the original GPS position.

Our proposed method shares some of the concepts explained above to increase the distinctiveness of visual feature descriptors. Instead of using a coarse sensor pose to project the reference model into the camera image, as in (Reitmayr and Drummond, 2007), we use it to project the camera features onto a coarse model of the environment. Thereby, we gain their ASPAC comprising of the coarse 3D position, absolute scale, and absolute orientation, making it possible to constrain the set of reference features to match against in 3D space. This not only makes it easier to account for the different accuracies of the different sensor readings, but more importantly, the set of reference features to match against is determined for every camera feature individually. This then makes it possible to deal with repetitive visual features. While all features corresponding to windows on a building façade would fall into the same orientation bin, and most likely the same position bin (according to the partitioning proposed by (Arth et al., 2012)), our proposed method can help distinguishing them.

2.2 Evaluation Methods and Ground Truth Datasets

The most reliable way of evaluating a localization method is to compare its results with ground truth. However, it is generally a tedious task to determine the ground truth pose for real camera images – particularly in wide area outdoor environments.

(Irschara et al., 2009) do not have ground truth information and therefore measure the effective number of inliers to rate if a localization succeeded or not. (Ventura and Höllerer, 2012) synthesize camera images as unwarped parts of omnidirectional images. For ground truth, they use the position of the omnidirectional camera determined in the SfM process to create the reference map. Similarly, (Arth et al., 2012) simulate online-created panoramic images as subsets of existing full panoramas for evaluation, and manually set the corresponding ground truth position. There are datasets of real handheld camera images with corresponding 6DoF ground truth poses, but these either only contain planar tracking templates (Lieberknecht et al., 2009) or 3D objects captured indoors (Sturm et al., 2012) and without any associated sensor readings. Existing datasets for wide area outdoor environments exist in the robotics research domain, e.g. (Wulf et al., 2007), and consequently do not contain handheld camera motion.

The dataset we explain in this paper has recently been published as a poster (Kurz et al., 2013) and is publicly available for research purposes[1]. It contains sequences of an urban outdoor environment taken with an off-the-shelf mobile phone, which include the readings of GPS, compass and the direction of gravity. Most importantly, it comprises of ground truth information on the geometry and texture of the environment, and the full 6DoF ground truth camera pose for every single frame.

3 PROPOSED METHOD

We propose a visual 6DoF localization framework that – in addition to the camera image – employs the auxiliary sensors, which off-the-shelf smartphones are equipped with, to estimate an accurate camera pose. A GPS receiver provides a coarse absolute position, an electronic compass measures the device's heading and the direction of gravity is obtained from inertial sensors. Together with the assumption that the device is approximately 1.6 meters above the ground floor, a coarse 6DoF camera pose can be computed. While this pose usually is not accurate enough for precisely registered visual augmentations in the camera image, we describe how it can be used to support computer vision methods that enable a more accurate camera pose estimation.

Our work is based on a state-of-the-art visual localization and tracking framework, using local image features and 2D-3D point correspondences. In this paper, we make contributions to advance state-of-the-art in feature detection, feature description, and feature matching for (wide-area) outdoor applications by giving features an Absolute Spatial Context, which will be explained in the following.

[1] http://www.metaio.com/research

3.1 Required Environment Model

As this work aims to localize a camera in a known environment, we require a model that describes the environment in a way that enables determining correspondences between the model and parts of a camera image. Our method is based upon a sparse representation of the environment comprising of 3D points with associated feature descriptors that will be explained in more detail in 3.3. Such a kind of model, which we will refer to as *reference feature map*, can be obtained by means of structure from motion methods, e.g. (Arth et al., 2012), or from synthetic views of dense environment model as explained in section 5.1. We use a custom feature descriptor based on a similar approach to SIFT (Lowe, 2004), but optimized to perform in real-time on mobile devices. This descriptor uses the direction of gravity to normalize an image patch around a feature before its description, as proposed in (Kurz and Benhimane, 2011). The reference feature map describes local parts of the environment in great detail, but does not contain any topological or global information.

Additionally, our method requires a coarse but dense polygonal representation of the environment's surfaces. Such models can, for example, be obtained by extruding floor plans. This representation will be referred to as *reference surface model*, and neither needs to contain any details nor does it need to be accurately registered. As it will be described in the following, it is only used to aid the process of feature detection, description and matching, but it's coordinates are never used for pose estimation.

Figure 1 shows on the left the reference feature maps of four building façades in different colors and the reference surface models in black. The surface models are stored as a set of 3D triangles and the reference feature maps are stored as sets of 3D points with associated ASPAC-aware feature descriptors. Since both models do not share any 3D points, they are stored separately.

All models are required to be in a consistent and geo-referenced coordinate system. We assume such data can be made available for the majority of cities soon.

3.2 Environment Model-guided Feature Detection

The detection of image features is commonly used to speed up finding correspondences in images containing the same object or scene from different views. Instead of comparing patches around every pixel, comparison and matching is only performed for salient

Figure 2: Comparison of regular feature detection on the entire image (left) and the proposed environment model-guided approach (right).

image features. It is crucial to the whole process of describing and matching features for camera localization, that the detected features are well distributed and that many of them actually correspond to the object or environment our reference model describes.

In particular in outdoor environments, large parts of the camera image often contain objects which are not part of the model. Examples include clouds in the sky, the floor, trees, cars and pedestrians potentially occluding parts of the model, see figure 2 left. Therefore, we developed a method to make the feature detection process focus on the parts of the image that most likely correspond to something meaningful for localization.

As described above, we use GPS, compass and inertial sensors to compute a coarse 6DoF pose of the camera in a global coordinate system, which we will refer to as *sensor pose*. To account for inaccurate GPS, we make sure the pose is not located inside the surface model and not facing surfaces too close to the camera. To this end, we push the pose backwards along the principal axis until the closest intersection of this axis and the surface model is at least 15 meters away from the camera.

Based on this pose and the known intrinsic camera parameters, we project the reference surface model into the camera image. The resulting mask is used to only extract features in the camera image where parts of the model project into the image. Additionally, we propose to not extract any features in the lower hemisphere centered around the camera, because the parts of the model located below the horizon line are often occluded by pedestrians or cars.

Figure 2 compares the distribution of extracted features from two images using a regular approach (left) with our proposed method (right).

In all cases, we use the FAST corner detector (Rosten and Drummond, 2006) and find a threshold that results in 300 corner features. It is apparent that our proposed method leads to a significantly higher ratio of detected features, which correspond to parts of the environment model, than in the regular approach. As a result, the robustness against background clutter and partial occlusions of the environment is increased. In section 5.2 we will show that this also results in significantly increased localization success rates.

3.3 Absolute Spatial Context-aware Visual Feature Description

Given a coarse reference surface model and a coarse sensor pose of the camera, we are not only able to determine which pixels of the camera image most likely contain parts of the model, but we also retrieve a coarse position of the 3D point $\mathbf{P}(u,v)$ corresponding to those pixels. The position can for example be obtained by ray casting or by rendering the surface model into a position map. Additionally, the sensor pose provides every feature with an absolute orientation. In the following, we will describe how this information can be used to improve feature description by giving features in the camera image an Absolute Spatial Context (ASPAC).

As stated earlier, an important challenge for visual outdoor localization in urban environments is to deal with repetitive visual features. Not only do the four corners of a window look the same (except for their global orientation, which is part of their Absolute Spatial Context), but there are also multiple windows on a façade side by side and on top of each other that look exactly the same. Additionally, man-made environments tend to contain visually similar features at different physical scales. It is crucial for any visual camera localization method to distinguish these repetitive features to be able to determine an accurate camera pose.

In this work, we use the term *Absolute Spatial Context* to describe the absolute scale, the absolute position, and the absolute orientation of a feature. As opposed to the common definition of a feature's scale, position and orientation, which are defined in the (2D) coordinate system of the camera image, the Absolute Spatial Context is defined in a (3D) global world coordinate system.

Awareness of Absolute Scale makes it possible to distinguish features with similar visual appearance at different physical scales. Most state-of-the-art camera localization and tracking methods based on local image features are scale-invariant. A common way to make feature detection and description invariant to scale, which is also used in our approach, is to use image pyramids that represent a camera image at different scales. This makes it possible to detect and describe visual features of an object in a similar way no matter if it is 1 meter away from the camera or 5 meters away.

As this scale invariance happens in a projected space, i.e. the camera image, it is impossible to distinguish scale resulting from the distance of an object to the camera from the actual physical scale of an object. Invariance to scale resulting from the distance of the camera to an object is clearly desirable in many applications, and was the original motivation for scale-invariance. However, in the presence of similar features at different physical scales, invariance to scale makes them indistinguishable.

In the following, we will use the term *feature scale* as a scalar value describing the width and height of the squarish support region of the feature's descriptor. Given the coarse sensor pose and the coarse 3D position $\mathbf{P}(u,v)$ of a visual feature located in the camera image at pixel (u,v), the distance from the optical center of the camera to the feature point $d(u,v)$ can be easily computed.

Based on this distance, the intrinsic camera parameters, and the scale of a feature in pixels s_{pix} as described above, we propose to compute an approximation of its absolute physical scale $s(u,v)$ as

$$s(u,v) = s_{\text{pix}}(u,v)\frac{d(u,v)}{f}, \tag{1}$$

where f is the camera's focal length. This absolute physical scale is computed and stored for every feature.

Awareness of Absolute Position can help distinguishing between repetitive features that are located at different positions, such as similar windows on a building façade. To this end, we simply store the approximate 3D position of a feature computed from the sensor pose and the reference surface model as

$$\mathbf{p}(u,v) = \mathbf{P}(u,v). \tag{2}$$

This position is, of course, inaccurate, but we will discuss in 3.4 how it can aid and speed up the process of feature matching.

Awareness of Absolute Orientation makes similar features at different absolute orientations distinguishable, as described in (Kurz and Benhimane, 2011). We use gravity-aligned feature descriptors (GAFD) that take the measured direction of gravity projected

into the camera image as feature orientation. This orientation is then used to normalize an image patch around a feature before description. Thereby, repetitive features at different orientations, e.g. the four corners of a window, are described in a distinct way while the description is still invariant to the orientation of the camera. We denote the visual descriptor, which is based on a histogram of gradient orientations in the image patch, as $\mathbf{v}(u,v)$.

Additionally, we compute and store the dominant gradient direction o_{gradient} in a patch around the feature relative to the orientation of the gravity o_{gravity} as an additional part of the Absolute Spatial Context.

$$o(u,v) = \left| o_{\text{gradient}} - o_{\text{gravity}} \right|_{\text{angle}} \tag{3}$$

$$|\alpha|_{\text{angle}} = \begin{cases} \alpha + 2\pi, & \text{if } \alpha \leq -\pi \\ \alpha - 2\pi, & \text{if } \alpha \geq \pi \\ \alpha, & \text{else.} \end{cases} \tag{4}$$

Figure 3 plots an exemplary distribution of absolute orientations of the features located on a building façade. We clearly observe peaks at all multiples of $\pi/2$, i.e. 90 degrees, which are very common in man-made environments.

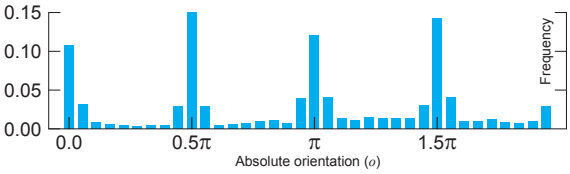

Figure 3: Distribution of the absolute orientation of features on a building façade comprising of mainly horizontal and vertical structures.

All the properties of a feature described above remain constant for varying camera positions and orientations. Therefore, they can be used to add distinctiveness to visually similar features, as they are very frequent in man-made environments. Additionally, they can be used to significantly speed up feature matching, which will be explained in the following.

3.4 Matching Absolute Spatial Context-aware Features

After detecting and describing features from a camera image, the matching stage is responsible for determining correspondences between these camera features and the reference feature map. We do this by finding for every camera feature the reference feature with the lowest dissimilarity using exhaustive search. Based on the resulting 2D-3D correspondences, the 6DoF pose of the camera can finally be determined.

In this paper, we propose to use the Absolute Spatial Context of visual features to speed up the matching process by precluding potential matches where the context is not consistent. Thereby, for a majority of combinations of camera and reference features, the expensive step of computing the distance of their visual descriptors can be skipped. This not only results in faster matching, but also provides more correct matches, because the Absolute Spatial Context prevents similar looking features that differ significantly in their global scale, position, or orientation, to be matched.

As described in the previous section, our feature description $\mathbf{d}$ is composed of

- the absolute position $\mathbf{p}$,
- the absolute scale s,
- the absolute dominant gradient orientation with respect to gravity o, and
- and a gravity-aligned visual feature descriptor $\mathbf{v}$.

To compute the dissimilarity of two features, our method computes intermediate distances δ_i followed by a check if these intermediate distances are below given thresholds. If this condition is fulfilled, the next intermediate distance is computed. Otherwise, the dissimilarity of the two features is set to infinity without any further computations.

The dissimilarity is defined as

$$\left\| \mathbf{d}_i - \mathbf{d}_j \right\| = \Delta_1 \tag{5}$$

where $\Delta_1, \Delta_2, \Delta_3$, and Δ_4 are defined as

$$\Delta_k = \begin{cases} \infty, & \text{if } \delta_k >= t_k \\ \Delta_{k+1}, & \text{if } \delta_k < t_k. \end{cases} \tag{6}$$

and

$$\Delta_5 = \delta_5. \tag{7}$$

As intermediate distances we use the distance on the x-y plane

$$\delta_1(\mathbf{d}_i, \mathbf{d}_j) = \left\| (\mathbf{p}_i - \mathbf{p}_j)\,[1,1,0]^\top \right\|, \tag{8}$$

the distance along the z axis (i.e. vertical)

$$\delta_2(\mathbf{d}_i, \mathbf{d}_j) = \left\| (\mathbf{p}_i - \mathbf{p}_j)\,[0,0,1]^\top \right\|, \tag{9}$$

the ratio of absolute scale

$$\delta_3(\mathbf{d}_i, \mathbf{d}_j) = \max(s_i, s_j) / \min(s_i, s_j), \tag{10}$$

the difference in absolute orientation

$$\delta_4(\mathbf{d}_i, \mathbf{d}_j) = \left| \left| (o_i - o_j) \right|_{\text{angle}} \right|, \tag{11}$$

and the visual descriptor distance

$$\delta_5(\mathbf{d}_i, \mathbf{d}_j) = \left\| (\mathbf{v}_i - \mathbf{v}_j) \right\|. \tag{12}$$

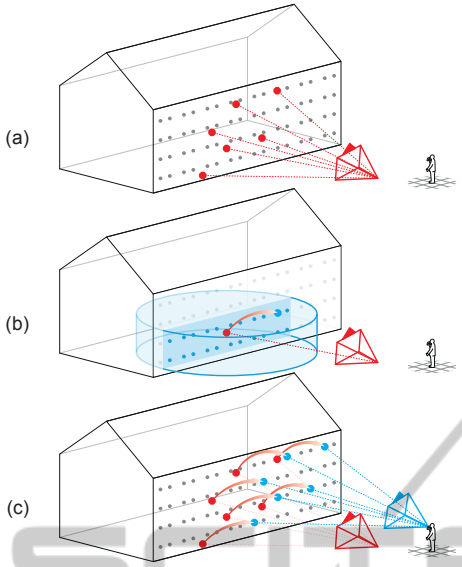

Figure 4: Determining the coarse absolute position of camera features (as part of their Absolute Spatial Context) based on the sensor pose (a) enables constraining feature matching to reference features that are located within a cylinder centered at the coarse position (b). Finally, the accurate camera pose can be determined based on a set of correct matches between 2D camera features and 3D reference features (c).

Note, that we treat the spatial distance along the z-axis, i.e. the altitude, differently than the distance along the other two axes. This is based on the assumption that the altitude is the most reliable part of the determined 3D position of a camera feature, because it is less heavily affected by an inaccurate compass heading or GPS position, as is also shown in figure 4.

The benefit of the proposed method to project camera features into the environment over projecting the reference feature map into the coordinate system of the camera, is twofold. Firstly, depth is preserved and avoids matching camera features against reference features that are occluded or at the backside of a building. Secondly, the transformation into a different coordinate system and the computation of feature properties in this coordinate system, which is performed in every frame during localization, is in our case only done for hundreds of camera features, instead of tens of thousands of reference features.

3.5 6DoF Localization Framework

A flow diagram of our proposed framework for 6DoF camera localization on mobile devices is shown in figure 5. It combines the above steps to establish correspondences between features in the camera image and the reference model with a pose estimation functionality. Every live frame consists of a camera image and a set of sensor readings measured at a time close

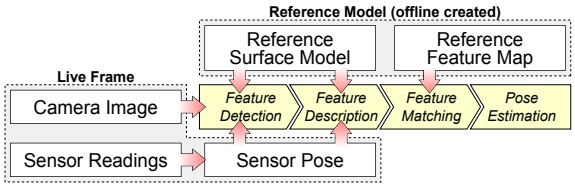

Figure 5: Flowchart of the proposed camera localization framework for outdoor environments.

to when the image was taken. Based on these sensor readings, we first compute a coarse sensor pose, which together with the reference surface model is then used in the feature detection stage, as described in section 3.2.

Afterwards, the features are described as specified in section 3.3, which again requires the sensor pose and the surface model to determine the Absolute Spatial Context (ASPAC) of the features. Eventually the features of a live camera image are matched against those of the reference feature map, according to the method explained in section 3.4. The resulting correspondences serve as a basis for the pose estimation step, which performs PROSAC (Chum and Matas, 2005) followed by a non-linear pose optimization based on all inlier matches.

In an outdoor handheld AR application, this localization step would be followed by frame-to-frame tracking. This paper, however, focuses on camera localization (i.e. initialization) only.

4 OUTDOOR 6DoF GROUND TRUTH DATASET

A quantitative evaluation of a 6DoF localization framework requires ground truth information. That is, for a set of given (realistic) input data, i.e. camera image, camera intrinsics, all sensor readings, and the required reference models, we need the expected ground truth output data, i.e. a 6DoF pose. As discussed in 2.2, there are ground truth datasets for visual localization and tracking available, but none of them fulfills our requirements to evaluate the framework proposed in this paper.

In the following, we describe our extensive procedure to create the first ground truth dataset for outdoor 6DoF handheld camera localization comprising of:

- a highly accurate, geo-referenced, and textured 3D model of a real urban environment spanning approximately 10,000 square meters,
- video sequences containing over 45,000 individual images of the environment with realistic handheld camera motion taken from different locations

with an off-the-shelf mobile phone,

- the sensor readings of GPS, compass, and the gravity vector for each image of the sequences mentioned above, and
- a very accurate 6DoF ground truth pose for every single camera image.

One important aspect of the design of this dataset is that it allows for easy expansion by adding more sequences taken with different devices, from different users, and under different illumination.

4.1 Model Acquisition

We chose an office park as a testing environment, which comprises of a large parking lot, different buildings, some parking lanes, and small streets. The covered area is approximately 100 by 100 meters wide.

As we aim to create a very precise and detailed model of the environment, SfM methods that reconstruct a sparse point cloud based on a multitude of images taken from different positions, are not suitable in this case. Instead, we used a FARO Focus 3D laser scanner to create nine, high-precision panoramic laser scans with texture information for different parts of the environment. The individual scans were then registered to a common coordinate system using proprietary software based on 3D-3D correspondences of registration spheres that were placed in the environment. The merged model has finally been georeferenced based on the latitude and longitude of a set of building corners obtained from OpenStreetMap[2].

As a result, we obtain a highly precise, dense and textured environment model which is referenced with respect to a global world coordinate system. The full registered model with color information is shown in figure 7, rendered from a bird's-eye view.

4.2 Sequence Recording at Known Camera Positions

There are two important aspects to keep in mind when recording sequences for testing. These are *relevance* for the targeted application and *universality*. It is important to use a capturing device and camera motions similar to those that can be expected to be used in real applications. It is crucial, that the dataset comprises of a high variance in parameters, such as the camera position, for the data to be considered universal and representative.

We decided to use an iPhone 4 mobile phone because it is a very common device and allows to obtain the GPS position, compass heading and a measurement of the gravity vector for every image. The image resolution is set to (480 $\times$ 360) pixels and the camera's intrinsic parameters were calibrated offline using a checkerboard pattern and Zhang's method (Zhang, 2000). We use a custom-made application to capture image sequences with all relevant sensor readings at a frame rate of $\sim$25 Hz and save them to files.

To ensure a universal and representative set of camera sequences, it is important to cover many different camera positions distributed over the entire test area. As this area spans about 10,000 square meters, installing an external tracking system that measures the 6DoF pose of the phone with a precision that can be considered ground truth is very complex, if not impossible. Therefore, we limit ourselves to a set of 156 discrete camera positions, which are spread all over the area and are chosen such that placing a camera to these positions is easy to achieve.

Our test environment comprises of a large parking lot and two smaller parking lanes, which are divided into individual cells by white markings on the ground. These serve as constant markers, since they are unlikely to change in the near future. We use the crossings and end points of these markings as survey points, and measure their precise 3D positions with a total station (Trimble 3603 DR). Based on 3D-3D correspondences between additionally measured points on the buildings, and the corresponding points in our ground truth model, the survey points are finally converted into the common world coordinate system. Figure 7 displays these survey points as red circles.

We then divide the process of obtaining sequences with 6DoF ground truth poses into two steps. In the first step we capture sequences at known 3D camera positions and recover the corresponding 3DoF orientation in a second step. Attaching a lead weighted string of known length to the phone's camera, makes it easy to precisely move the device to a known 3D position. As shown in figure 6, we hold the mobile phone directly over a survey point $\mathbf{s}_i$ on the ground with a string of known length h_i to be sure the camera is located at $\mathbf{t}_i$, which can simply be computed as

$$\mathbf{t}_i = \mathbf{s}_i + h_i \cdot [0,0,1]^\top. \tag{13}$$

We use strings at lengths of 1 and 1.8 meters, which can be considered to represent the actual heights users hold their mobile devices. An interesting property when taking sequences at a height of 1 meter, is that they contain much more occlusions of the buildings because of the cars on the parking lot. While capturing and recording sequences (some frames of two exemplary sequences are shown in fig-

[2]http://www.openstreetmap.com

Figure 6: Sequence recording at a known camera position using a lead weighted string and survey points on the ground.

ure 6), the camera only undergoes rotational movements and does not change its position. Since (Chittaro and Burigat, 2005) found out that users prefer standing while using the screen of a mobile phone for information, we believe that our sequences have realistic kinds of camera motion for handheld Augmented Reality applications.

4.3 6DoF Ground Truth Recovery

For the second step of the ground truth acquisition process, we prepared an edge model of the environment based on the ground truth model. Using the coarse camera orientation obtained from the sensor readings and the accurately known 3D position of the camera, we project the edges into the camera image. We then find the orientation for which the model best fits gradients in the camera image using exhaustive search in a neighborhood around the initial orientation estimate. Finally, the recovered 3DoF camera orientation, together with the 3DoF known ground truth position, make the 6DoF ground truth pose. To account for potential errors in labeling or recovery of the rotation, the ground truth poses of all images have been manually verified by rendering a wireframe model onto the video stream. Figure 7 displays the recovered 6DoF ground truth pose for three exemplary images of the dataset, in green.

In total, we recorded 100 sequences from different locations and heights imaging façade1 and an additional 25 sequences of façade4, cf. figure 1. All sequences comprise over 45,000 images and the corresponding sensor information. For every frame, we recovered the 6DoF ground truth pose. Since we used parking markings as easily identifiable camera locations, it was convenient to record sequences at different times of the day and under varying weather conditions. In future the database can be easily expanded by more sequences comprising of more drastic weather changes, e.g. snow or rain, or to contain data from other devices and cameras.

5 EVALUATION AND RESULTS

In the following, we evaluate our proposed localization framework, described in section 3, using the outdoor 6DoF ground truth dataset, which was explained in the previous section.

5.1 Ground Truth Localization Test

In order to make use of the detailed ground truth model of the environment, we first need to convert it into the model representation required by our method as described in section 3.1. We define the four different building façades, shown color-coded in figure 1 (left), as objects we are interested in for localization while the rest of the environment – mainly consisting of the floor, cars and trees – is not relevant. For each of these objects, we create a localization reference model as follows.

By rendering the ground truth model from different virtual viewpoints, we gain a set of synthetic photo realistic views of the environment, where for every pixel the corresponding 3D position is known. We then detect features with known 3D coordinates from these views and describe them as elaborated in 3.3. Note, that instead of using a coarse pose computed from GPS, compass and inertial sensors, we use the precisely known pose of the virtual camera used to render the view to provide the Absolute Spatial Context. Finally, we determine out of the descriptors from all the views, a representative feature descriptor set comprising of 2,000 features per object, as explained in (Kurz et al., 2012). This set of descriptors and features is then used as a reference feature map. The reference surface models have been manually created for this test and are shown in figure 1 (left).

We run all tests offline on a PC, but use the same localization framework that runs in real-time on mobile devices.

As we are interested in localization (or initialization) only, and not tracking, we treat every single

frame individually. The image and the sensor readings (GPS position, compass heading, and gravity vector) are read from files and provided to the system. We then perform the whole localization pipeline (as explained in 3.5) on this data as if it was live data. Eventually, the framework either returns a determined camera pose or replies that it did not succeed to localize the camera.

In video-see-through Augmented Reality applications, it is most important that the visualization (rendered with the obtained pose) appears correctly registered with the camera image. Therefore, we use the average re-projection error e of a set of 3D vertices located on the reference model as error measure. This error can be computed as

$$e = \frac{1}{k}\sum_{i=1}^{k} \|\mathbf{K}(\mathbf{R}_{obt}\mathbf{t}_{obt})\, v_i - \mathbf{K}(\mathbf{R}_{gt}\mathbf{t}_{gt})\, v_i\| \qquad (14)$$

where v_i are the 3D vertices, $\mathbf{K}$ denotes the camera intrinsic matrix, and $(\mathbf{R}_{obt}\mathbf{t}_{obt})$ and $(\mathbf{R}_{gt}\mathbf{t}_{gt})$ are the obtained pose and ground truth pose respectively. We require the re-projection error to be less than a threshold of 4 pixels for the pose to be considered correct.

The framework is run on all captured frames in three different configurations:

Naïve. A naïve approach, where feature detection is performed on the entire image, feature description uses GAFD, and the matching only compares the visual descriptors.

Orientation. Similar to the naïve approach but with orientation-aware feature matching that only compares features with a similar heading analog to what is proposed in (Arth et al., 2012).

Proposed. Our proposed method, where feature detection, feature description, and feature matching make use of the ASPAC provided by the sensor values.

To evaluate how the individual approaches scale with an increasing reference model, we evaluate all sequences in all configurations with two different reference models:

Only. Only uses the reference model of the building façade, which is imaged in the current sequence.

All. All four reference models shown in figure 1 on the left are combined to a large reference model.

We chose the following thresholds in our evaluation:

$t_1 = 10{,}000$ mm – Spatial distance on the x-y plane.

$t_2 = 2{,}000$ mm – Spatial distance along the z axis.

$t_3 = 1.3$ – Ratio of absolute scale.

$t_4 = 120^\circ$ – Difference in absolute orientation.

The reason for t_4 being large, is that the scene mainly consist of windows, and the corners of these windows, which provide a majority of features, usually have at least two orthogonal dominant gradient directions in their neighborhood making the absolute orientation an unreliable parameter in this environment. The vertical distance threshold t_2 was chosen as two meters, to ensure discrimination between the windows of different building stories, which are usually about 3 meters high.

In the *Orientation* approach, we use a threshold of $\pm 30^\circ$ as in the original paper.

5.2 Test Results

First of all, we evaluate for all configurations the ratio of the frames in our ground truth dataset for which the localization framework determines a correct pose. The results are given in table 1, and show that the *Orientation* approach performs better than the *Naïve* approach on the large reference model (All). When dealing with only one façade (Only), the *Naïve* provides better results than *Orientation*. In this case, the *Orientation* approach seems to preclude correct matches due to inaccurate compass heading values.

Our *Proposed* method clearly outperforms all other methods in terms of correctly localized frames. It also is apparent that our *Proposed* method scales very well with an increasing reference model. Scaling the number of reference features by a factor of four (Only → All) has a minimal effect, while the ratio of correctly localized frames drops significantly for the *Naïve* approach.

Table 1: Ratio of correctly localized frames in the outdoor ground truth dataset.

Sequences\Method	Naïve	Orient.	Proposed
Façade1 (Only)	30.87%	22.91%	50.03%
Façade4 (Only)	4.98%	3.66%	9.00%
Total (Only)	25.54%	18.95%	**41.58%**
Façade1 (All)	16.20%	20.99%	49.98%
Façade4 (All)	2.80%	3.01%	9.00%
Total (All)	13.44%	17.29%	**41.54%**

The absolute numbers given above might appear low compared to the results of other papers (e.g. (Arth et al., 2012)(Ventura and Höllerer, 2012)). It is important to keep in mind that the dataset we use is realistic, and therefore, particularly hard compared to tests in the literature. All reference feature maps are based on the panoramic images from the laser scanner while the test sequences were taken with a mobile phone at different days and weather conditions. Additionally, the scene – particularly façade4 – contains a sig-

nificant percentage of repetitive visual features, which are mainly windows that additionally reflect the sky, resulting in frequent changes in their appearance. Another challenge is that the majority of our sequences contain cars partially occluding the building façades.

Our proposed method deals well with repetitive visual features, but still has problems with significant changes in illumination. For some of the sequences in the dataset, not a single frame was localized correctly with any method simply because the illumination is too different from that in the reference model. Here, further research on algorithms to compute the visual descriptor in a fashion invariant to illumination is needed.

Table 2: Impact of the individual proposed steps and intermediate distances on localization cost and quality in *Façade1* (Only).

Distance\Measurement	Computed δ_5	Correct
(a) $\Delta_1 \leftarrow \delta_5$ (Naïve)	100.00%	30.87%
(b) $\Delta_1 \leftarrow \delta_5$ + EMGFD	100.00%	43.56%
(c) $\Delta_2 \leftarrow \delta_5$	43.14%	46.92%
(d) $\Delta_3 \leftarrow \delta_5$	12.89%	47.91%
(e) $\Delta_4 \leftarrow \delta_5$	2.86%	**50.39**%
(f) $\Delta_5 \leftarrow \delta_5$ (Proposed)	**2.16**%	50.03%

To evaluate the impact of the steps involved in our proposed method and the intermediate distances computed in the matching stage, we repeated the experiment above in more different configurations. Beginning from the naïve approach, every row in table 2 adds one more of the steps that we proposed before finally computing the visual descriptor distance. Starting from environment model-guided feature detection (denoted by EMGFD), we added the constraint on the distance on the x-y plane, the distance along the z axis, the ratio of absolute scale, and the difference in absolute orientation. We observe that the first four steps of our proposed method (b,c,d,e) result in a continuously increased ratio of correctly determined poses, while the number of expensive comparisons of visual feature descriptors (δ_5) needed decreases monotonically. The configuration (e) localizes over 1.6 times as many frames correctly as the naïve approach and requires less than 3% of the visual descriptor comparisons.

Adding the constraint on the absolute feature orientation (f) results in even less visual descriptors being compared (factor 0.76) but also slightly decreases the ratio of correctly localized frames (factor 0.99). Therefore, depending on the application, it can make sense to omit this step because the absolute feature orientation already contributed to the distinctiveness of the (gravity-aligned) visual descriptors.

6 CONCLUSIONS AND FUTURE WORK

This paper presented a framework for visual camera localization that utilizes the sensors modern mobile phones are equipped with to provide local visual feature descriptors with an Absolute Spatial Context (ASPAC). This novel feature description method overcomes visual repetitiveness in urban environments, which is one of the most pressing problems for visual camera localization. Moreover, we presented the first publicly available dataset comprising of real camera sequences and sensor readings captured outdoors using a mobile phone with accompanied 6DoF ground truth poses. Using this comprehensive dataset, we showed that our proposed method clearly outperforms a naïve approach using only the direction of the gravity and visual information, and an approach similar to a recently published work (Arth et al., 2012) that additionally uses the heading orientation to constrain feature matching.

Our proposed method to detect, describe and match features shows that the auxiliary sensors of mobile devices can help to not only get better localization results, but to also speed up matching by precluding the comparison of visual descriptors of features with a largely different Absolute Spatial Context. Additionally, the proposed scheme can be very well applied to feature matching in hardware, which will make matching against large databases of reference feature maps virtually free of cost in the future.

While sensor values are clearly helpful as long as they are reasonably accurate, our proposed method will not work if some of the sensor readings are very imprecise. For these cases, we are currently looking into fallback strategies, e.g. by switching between our proposed method and a naïve method in every other frame if localization does not succeed for a certain period of time.

The established outdoor 6DoF ground truth dataset is not dependent on the proposed localization framework but can be used to evaluate any localization method including those relying on color features, edge features, or even model-based approaches that require a dense and textured reference model. Furthermore, the dataset enables benchmarking frame-to-frame tracking methods as it comprises image sequences captured with a handheld camera with different levels of realistic interframe displacement. As capturing additional sequences does not require any hardware, except a capturing device and a lead weighted string, we plan to expand the dataset by more sequences taken with different devices, from different users, and during different weather.

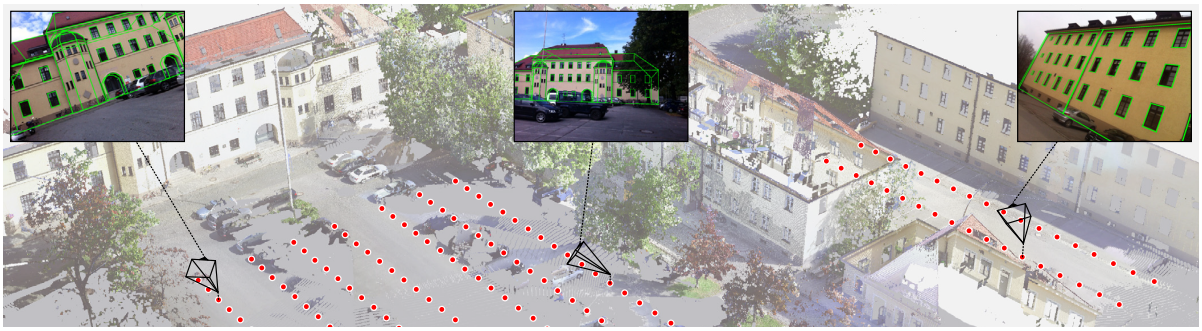

Figure 7: Our outdoor ground truth dataset comprises of a precise 3D model of the environment and over 45,000 camera images with sensor readings and 6DoF ground truth poses. Exemplary images are shown as insets with their ground truth poses rendered as frustra.

ACKNOWLEDGEMENTS

This work was supported in part by the German Federal Ministry of Education and Research (BMBF, reference number 16SV5745, PASSAge) and the German Federal Ministry of Economics and Technology (BMWi, reference number 01MS11020A, CRUMBS). The authors further wish to thank Darko Stanimirović and Marion März for their help on the ground truth dataset and FARO Europe for providing us with the laser scans.

REFERENCES

Arth, C., Mulloni, A., and Schmalstieg, D. (2012). Exploiting Sensors on Mobile Phones to Improve Wide-Area Localization. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*.

Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., and Pollefeys, M. (2012). Leveraging 3d city models for rotation invariant place-of-interest recognition. *Int. Journal of Computer Vision (IJCV)*, 96(3):315–334.

Chittaro, L. and Burigat, S. (2005). Augmenting audio messages with visual directions in mobile guides: an evaluation of three approaches. In *Proc. Int. Conf. on Human Computer Interaction with Mobile Devices and Services (Mobile HCI)*.

Chum, O. and Matas, J. (2005). Matching with PROSAC - Progressive Sample Consensus. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Fritz, M., Saenko, K., and Darrell, T. (2010). Size matters: Metric visual search constraints from monocular metadata. In *Advances in Neural Information Processing Systems (NIPS)*.

Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*.

Knopp, J., Sivic, J., and Pajdla, T. (2010). Avoiding confusing features in place recognition. In *Proc. European Conf. on Computer Vision (ECCV)*.

Kurz, D. and Benhimane, S. (2011). Inertial sensor-aligned visual feature descriptors. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Kurz, D., Meier, P., Plopski, A., and Klinker, G. (2013). An Outdoor Ground Truth Evaluation Dataset for Sensor-Aided Visual Handheld Camera Localization. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*.

Kurz, D., Olszamowski, T., and Benhimane, S. (2012). Representative Feature Descriptor Sets for Robust Handheld Camera Localization. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*.

Lieberknecht, S., Benhimane, S., Meier, P., and Navab, N. (2009). A dataset and evaluation methodology for template-based tracking algorithms. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision (IJCV)*, 60(2):91–110.

Reitmayr, G. and Drummond, T. W. (2007). Initialisation for visual tracking in urban environments. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Proc. European Conf. on Computer Vision (ECCV)*.

Smith, E. R., Radke, R. J., and Stewart, C. V. (2012). Physical scale keypoints: Matching and registration for combined intensity/range images. *Int. Journal of Computer Vision (IJCV)*, 97(1):2–17.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proc. Int. Conf. on Intelligent Robot Systems (IROS)*.

Ventura, J. and Höllerer, T. (2012). Wide-area scene mapping for mobile visual tracking. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*.

Wulf, O., Nuchter, A., Hertzberg, J., and Wagner, B. (2007). Ground truth evaluation of large urban 6D SLAM. In *Proc. Int. Conf. on Intelligent Robot Systems (IROS)*.

Zhang, Z. (2000). A flexible new technique for camera calibration. *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(11):1330–1334.