2015 IEEE International Symposium on Mixed and Augmented Reality Workshops

Geolocation for Printed Maps Using Line Segment-Based SIFT-like Feature Matching

Gautier Minster*
Keio University

Guillaume Moreau†
École Centrale de Nantes

Hideo Saito‡
Keio University

ABSTRACT

This paper presents a method for the geolocation of printed maps. It enables the registration of unprepared maps with a Geographical Information System (GIS) database, and can for example be used as a first step to augment an unknown map.

We define and match local road pattern descriptors, which are similar to SIFT descriptors [6], but adapted to the case of simple textureless line segments. Using a processing pipeline commonly encountered in the feature-point based matching of texture images — composed of offline description and indexing, followed by an online description, matching and robust transformation estimation — we show that local descriptors can successfully register unprepared maps using only geographic features and no texture information.

Our method is scale and rotation invariant, and circumvents the two hurdles that are the level-of-detail, and the changing colormaps and textures, allowing the processing of large classes of printed maps.

Keywords: Printed map, geolocation, GIS, local descriptor, map registration.

Index Terms: I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Feature Representation I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Shape I.5.2 [Pattern Recognition]: Design Methodology—Feature evaluation and selection

1 INTRODUCTION

Printed maps are the traditional medium for the representation of geographic information, with ubiquitous use in tourism and urban planning, amongst other domains. Geographic Information Systems (GIS) based electronic maps are extensively used as well, yet do not make it easy to have natural interactions, such as collaboration between multiple users or annotations. Augmented maps [8] aim to tap into the vast resources of GIS, to dynamically overlay extra information on traditional printed maps.

Numerous methods have been proposed for this task, following the common pattern of feature extraction, GIS-registration, and finally tracking. These algorithms can be split in 3 categories, depending on the sort of maps they operate on: pre-prepared maps with added markers, unprepared maps with specific texture patterns, and arbitrary unprepared maps.

Methods based on markers require preparing the paper map beforehand: Reilly et al. [10] attached RFID tags to the back side, Schöning et al. [12] added visible ARToolkitPlus [13] markers around it, and Martedi et al. [7] added a dot pattern on top of it to perform point-pattern based tracking. SIFT [6] or SIFT-like texture features have been used by Reitmayr et al. [11] as well as Morrison et al. [8]. These methods only function on known maps, since the texture pattern changes arbitrarily between maps.

Finally, Yang et al. [14, 15] proposed registration methods based on the point-pattern of the road network intersections. Unlike the previous works, it enables the registration and tracking of arbitrary unprepared maps, under perspective views. The method lacks robustness to scale however, and its quadratic time complexity limits the size of the tracked areas.

In this paper, we propose a method for the registration of printed maps undergoing similarity transformations, using the line segment pattern of the road network. The algorithm transposes a texture matching framework similar to SIFT to textureless road networks. It can be used as a scalable first step to identify the area depicted in a printed map, before using one of the aforementioned works for realtime tracking and augmentation of a small area.

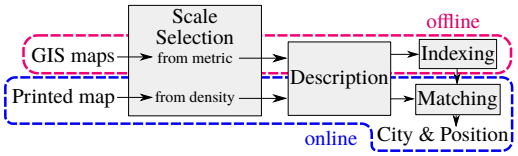


Figure 1: Overview of the algorithm

2 REGISTRATION ALGORITHM

The registration will operate on data from two sources: reference line segment data extracted from a GIS, and test line segment data extracted from the image of a printed map, for example using [1]. The algorithm is summarized in Figure 1.

2.1 Map representation

Both reference and test data are lists of segment endpoints. Since the reference segments may be expressed with inconveniently large and precise coordinates, we condition them to span a reasonable range for pixel images (e.g. a few thousand units in each dimension), and create an image representation of the segments at the chosen resolution.

The descriptor operates on such images, at varied scales. Large description areas would contain many pixels, so we iteratively downsample the image (by a factor 2), creating a pyramid. Multiple roads may overlap on one pixel in the pyramid, so we represent pixels as multi-sets (i.e. each set item has a counter): in the initial image, each pixel contains the roads that pass through it (all the counters equal 1); when downsampling, square blocks of pixels are merged by adding counters, thus giving roads spanning multiple pixels larger weights in the multi-sets.

2.2 Descriptor position and scale

Describing the local road pattern in a map region implies finding answers to several questions. Where are the distinctive road segment patterns? At what scale is a pattern relevant? How do we describe the pattern to make the description robust to the variations in different map sources?

*e-mail: gautier+keio@minster.io
†e-mail:guillaume.moreau@ec-nantes.fr
‡e-mail:hs@keio.jp

978-1-4673-8471-1/15 $31.00 © 2015 IEEE
DOI 10.1109/ISMARW.2015.24

We first explore the selection of description scales, followed by the positioning of descriptors, and finally the actual description method.

2.2.1 Scale selection

In the feature-point matching of texture images, keypoints can present perceptually important scales, such as the radius of a circle when the keypoint is its center. Keypoint detectors devise algorithms to determine such scales, using for example difference-of-Gaussians in scale-space [2, 5]. In the case of road segments however, this has no direct equivalent, and changes in the level-of-detail of the maps, combined with the potential fragmentation of even long roads, makes the problem nontrivial.

Instead of choosing the scales, we simply sample them: we determine, for each map to be described, a range of reasonable description scales, and use descriptor sizes in this range.

Metric information-based When dealing with data from a GIS, metric information is available: a range of scales expressed in world units is set (in the experiments, $100m$ to $500m$ of descriptor radius), and the corresponding pixel dimensions used.

Density-based For printed maps with unknown scales, no scale can be assumed *a priori*: depending on the image source (camera phone, scanned map, ...), and on the pictured region, we may process a small image with very concentrated roads, or a large image with spaced-out ones. We opt to compute a sort of pixel density, as a function of the region size: for a scale s (in pixels), let $G(s)$ be a square grid of cells of size $s \times s$. We define the density as follows:

$$\text{density}(s) = \frac{1}{t} \cdot \max_{C \in G(s)} \text{Card}(\{\text{non-empty pixels in } C\}) \quad (1)$$

where t is the total number of non-empty pixels in the image.

The range of description scales is determined by specifying a range of densities (in the experiments, 5% to 40%), and finding using binary-search scales s which best correspond to the two densities. Note that since the density is computed by taking the maximum value across cells of a grid, instead of iterating over all possible regions of size s, this is only an approximation of the real density, and as such may not even be monotonously increasing with s. It is, however, sufficient for our purpose.

2.2.2 Descriptor grid

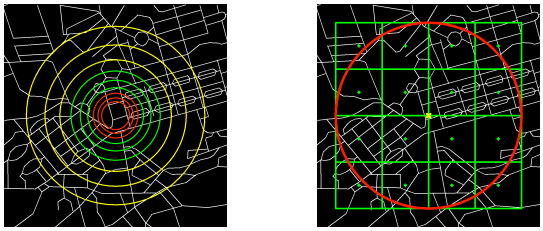
Lacking a reliable method to choose the locations of the descriptors, we instead opt to produce a dense sampling of the map, in both space and scale.

Let s_{min} and s_{max} be the extremal scales obtained from the scale selection process. We want to obtain a list of description radii to sample the range $[s_{min}, s_{max}]$. Each radius r should therefore be in $[s_{min}/2, s_{max}/2]$. Given a minimum description radius r_{min} (equal to 20 pixels in the experiments), the set of description radii is:

$$R = \left\{ r = r_{min} \cdot 2^{l+i/k}, \forall l \in \mathbb{N}, i \in [0,k[\,\middle|\, s_{min} \leq 2 \cdot r \leq s_{max} \right\} \quad (2)$$

where l specifies the octave (as defined in [6]), and $k \leq r_{min}$ is the number of intermediate descriptors per octave ($k = 3$ in the experiments). Each octave (i.e. admissible value of l) will correspond to a certain downsampling of the original map (as described in 2.1), and will be described using at most k different description radii. k is discussed in section 3.2, but r_{min} is merely a performance trade-off between large descriptor regions and large multi-sets that has little influence on the overall algorithm. Figure 2a shows an example of R, computed using density information for a printed map.

For each radius in R (the description scales/radii), we densely sample the map in the two space dimensions: the descriptors are computed over a square grid, of spacing equal to the description radius, such that an overlap exists between descriptor regions.



(a) Set R of description radii for a sample printed map ($k = 3$, $r_{min} = 20$). Each color represents a different scale octave. Note how the each doubling of the description radius involves a differently sampled multi-set segment image.

(b) A descriptor representation; in white, the road segments, in yellow, the region center, in red, the central Gaussian (represented only as a circle), and in green, the $N \times N$ histogram grid (assuming $a_{ref} = 0$)

Figure 2: Scale sampling and descriptor histogram grid

2.3 Local Road Pattern Descriptor

For each of the selected radii, at each grid position, in the appropriate downsampled map image, we describe the local pattern of road segments using histograms of segment orientations. The process is largely inspired by the SIFT [6] descriptor.

2.3.1 Descriptor orientation

To be able to match maps with arbitrary orientations, we aim to describe regions in a rotation-invariant manner. This is achieved by assigning each region a reference orientation, and computing the descriptor relative to that orientation.

The reference is obtained from a region-wide orientation histogram $H = (h_0, \ldots, h_{B-1})$ of B bins (in the experiments, $B = 9$), representing angles in $[0, \pi[$ (the segments are not oriented). The histogram is built by iterating over the pixels $(p_i)_i$ of the region, and computing the contribution of each segment in the pixel multi-sets. We weigh the contribution of a pixel p_i by a factor $w_g^{(i)}$ following a 2D Gaussian distribution centered on the region center, of standard deviation the region radius, to give more importance to center pixels. In p_i, a segment s_j with counter $c_j^{(i)}$ is weighted proportionally to the total size of the p_i multi-set. Let $h_{s_j}^+$ (resp. $h_{s_j}^-$) be the bin above (resp. below) the orientation of s_j, and $d_{s_j}^+$ (resp. $1 - d_{s_j}^+$) the distance in bin units between $h_{s_j}^+$ (resp. $h_{s_j}^-$) and the orientation. Summarizing, a segment s_j in a pixel p_i makes a contribution $W_j^{(i)}$:

$$W_j^{(i)} = w_g^{(i)} \cdot \frac{c_j^{(i)}}{\sum_k c_k^{(i)}} \cdot \begin{cases} 1 - d_{s_j}^+ & \text{to } h_{s_j}^+ \\ d_{s_j}^+ & \text{to } h_{s_j}^- \end{cases} \quad (3)$$

Once the global histogram is computed, the reference orientation is determined by finding the bin with largest value, and refining the estimated angle (since the bins can be fairly wide, 20° for $B = 9$) by fitting a 2nd order polynomial to the the maximal bin and its 2 neighbors. We thus achieve sub-bin accuracy for the reference orientation.

The obtained reference is an angle a_{ref} in $[0, \pi[$, which is not enough for a full rotation-invariance. Indeed, a rotation of the map by π leaves the global histogram unchanged, and consequently yields the same reference orientation. A solution is proposed in section 2.3.3, once all the descriptor data is computed.

2.3.2 Descriptor

The actual descriptor is an array of histograms, as shown in Figure 2b: the described region, rotated by the reference orientation, is divided into a $N \times N$ grid (in the experiments, $N = 4$), with one histogram per grid cell.

The computation of the histogram grid is essentially identical to the computation of the global histogram, with a minor addition: to avoid boundary effects (large, sudden changes in the descriptor when the road data varies slightly), the contribution of each pixel does not go solely to the closest histogram cell, but to the 4 closest cells, using bilinear interpolation to split the contribution.

The resulting descriptor is thus a set of N^2 histograms of size B.

2.3.3 π-rotation ambiguity

While the global orientation histogram is left unchanged by a π-rotation, the grid of histograms is not. This ambiguity can be resolved in different ways, for example by:

- avoiding the problem altogether by yielding 2 descriptors for each region, one oriented using a_{ref}, one using $a_{ref} + \pi$. This doubles the number of descriptors, which might be prohibitively expensive, given the dense sampling in space and scale that we use for map description.
- making the descriptor truly π-rotation invariant, by averaging the two possible descriptors. While the true invariance sounds appealing, the descriptor loses significant discriminative power.
- devising a method to choose between the two possibilities. This preserves the discriminative power and the number of descriptors, but depends heavily on the consistency of the choice method.

We choose this final option. The decision criterion we use is the following: for each half of the histogram grid, upper and lower, we sum the values of the 0-th bins (i.e. the bins corresponding to the reference angle), and compare them. We choose (arbitrarily) that the upper half should have the largest sum. If this is not the case after the computations of section 2.3.2, the grid undergoes a rotation by π, which simply translates to a reverse ordering of the cells in a flattened indexing of the grid.

The consistency of this method is discussed in Section 3.2 and Figure 4.

2.3.4 Capping and normalization

The obtained descriptor is finalized by a two-step process, again inspired from [6].

The first step aims at making the descriptor more descriptive in regions where a repetitive pattern dominates: the vector (i.e. the concatenation of the histograms from the grid) is normalized, and a cap c is applied to all its elements (in the experiments, $c = 0.15$). Experimentally, this helps make the lesser orientations stand out more, and have more influence on the distance between descriptors. This also helps match maps with different levels-of-details (the same road can be represented as multiple distinct segments, or a single one, depending on the map).

The second step is a re-normalization, to make descriptors comparable, and make a map where each segment is doubled (which can happen for separated highways and the like) have identical descriptors.

The final descriptor is a unit L_2-norm vector of $\mathbb{R}^{B \cdot N^2}$, which means that in the experiments, where $B = 9$ and $N = 4$, the descriptors are of size 144.

2.4 Matching and Geolocation

We now describe the offline indexing of reference maps, and online matching of test ones.

2.4.1 Offline description and indexing

The offline process is detailed in Algorithm 1. Reference maps $r \in \mathscr{M}_{ref}$ are described by densely sampling both scale and space as summarized in Function Describe, and an index is built from the descriptor data to allow for fast matching using nearest-neigbor queries. The experiments use the Fast Library for Approximate Nearest Neighbors, FLANN [9].

Function Describe(m, R_m): densely describe a map

Input: A map m, a set of radii R_m
Output: Set $\mathscr{D}$ of descriptors of m (position, scale, orientation, and vector data)

$\mathscr{D} \leftarrow \varnothing$
for r in R_m**:**
  $G_r \leftarrow$ 2D grid on m, of spacing r
  for Cell center p in G_r**:**
    $o, d \leftarrow$ `Descriptor`(m, p, r) (compute orientation o and descriptor data d)
    $\mathscr{D} \leftarrow \mathscr{D} \cup (p,\ r,\ o,\ d)$
return $\mathscr{D}$

Algorithm 1: Offline part of the registration (training)

Input: A set $\mathscr{M}_{ref}$ of reference maps, as lists of segments
Output: A FLANN index $\mathscr{F}$ of reference descriptors

$\mathscr{F} \leftarrow \varnothing$
for reference map r in $\mathscr{M}_{ref}$**:**
  $Img_r \leftarrow$ `MultiSetImage`(r)
  $\mathrm{R}_r \leftarrow$ `ScalesFromMetric`(Img_r)
  $\mathscr{D}_r \leftarrow$ `Describe` $(Img_r,\ \mathrm{R}_r)$
  `AddToIndex`$(\mathscr{F}, \mathscr{D}_r)$
return $\mathscr{F}$

2.4.2 Online registration

Algorithm 2 shows the online processing: the test printed map is densely described just like reference maps, the model estimated and refined, and the final identification performed.

Matching The matching process goes as follows: for each descriptor of the test map (of keypoint $k_t = (p_t, s_t, o_t, d_t)$, where p_t is the location, s_t the scale in pixels, o_t the orientation, and d_t the descriptor vector), the two nearest neighbors kp_{r1} and kp_{r2} in the index (i.e. the two most similar descriptors from the reference maps) are retrieved. Let $dist_1$ (resp. $dist_2$) be the distance between d_t and the descriptor of kp_{r1} (resp. kp_{r2}). We define a threshold t_{snn} (threshold second nearest neighbor, set to 0.98 in the experiments) on the ratio of $dist_1$ over $dist_2$, under which a match (kp_t, kp_{r1}) is kept.

The rationale behind this threshold is that test keypoints which may correspond to multiple reference keypoints are not very descriptive, and should therefore be discarded.

This process yields sets of raw matches $(\mathscr{R}_r)_{r \in \mathscr{M}_{ref}}$.

Random model estimation Transformation models between the test map and reference maps are estimated using the RANSAC [3] algorithm. More specifically, a random model fitting is repeated (up to a maximum number of iterations, $it_{max} = 2000$ in the experiments), in which a reference map is chosen using a weighted random choice, the weight of a reference map r being the proportion w_r of raw matches it produced:

$$w_r = \frac{\mathrm{Card}(\mathscr{R}_r)}{\sum_{u \in \mathscr{M}_{ref}} \mathrm{Card}(\mathscr{R}_u)} \tag{4}$$

Once the reference r has been chosen, a pair of matches from $\mathscr{R}_r$ is randomly selected, and a similarity (rotation, scaling, and translation, 4 degrees of freedom) is computed from the 2 matches.

Algorithm 2: Online part of the registration

Input: A list of segments representing a test printed map m, a FLANN index $\mathscr{F}$
Output: A list of plausible reference map matches, i.e. pairs $(C_r,\ M_r)_r$ of confidence indices and transformation models for each reference map r

$Img_m \leftarrow \texttt{MultiSetImage}(m)$
$\mathrm{R}_m \leftarrow \texttt{ScalesFromDensity}(Img_m)$
$\mathscr{D}_m \leftarrow \texttt{Describe}\,(Img_m,\ \mathrm{R}_m)$
$(\mathscr{R}_r)_r \leftarrow$ nearest-neigbor keypoint matches of $\mathscr{D}_m$ in $\mathscr{F}$, filtered by the second nearest-neigbor ratio threshold t_{snn}, grouped by reference map r
$(\tilde{I}_r, \tilde{M}_r)_r \leftarrow \texttt{RANSAC}((\mathscr{R}_r)_r)$ (inlier matches and transform model for each m_r)
$(I_r, M_r)_r \leftarrow \texttt{RefineModels}((\mathscr{R}_r, \tilde{I}_r)_r)$ (iterative refinement using least-squares estimations)
$(C_r)_r \leftarrow \texttt{Identify}((I_r, M_r)_r)$ (final decision making)
return $(C_r, M_r)_r$

Function RefineModels($(\mathscr{R}_r, \tilde{I}_r)_r$): refine initial estimates

Input: Raw matches, initial inliers $(\mathscr{R}_r, \tilde{I}_r)_r$
Output: Refined inliers and models $(I_r, M_r)_r$

for each reference map r**:**
  $I_r, M_r \leftarrow \tilde{I}_r$, **None**
  while $M_r =$ **None** or $I_r \neq \tilde{I}_r$**:**
    $\tilde{I}_r \leftarrow I_r$
    $M_r \leftarrow \texttt{WeightedLeastSquaresModelFit}(\tilde{I}_r)$
    $I_r \leftarrow \texttt{FindInliers}(\mathscr{R}_r, M_r, \varepsilon_a, \varepsilon_d)$
return $(I_r, M_r)_r$

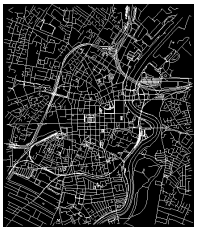
Using this estimated model, we iterate over the rest of the matches for r, and add those which fit the model to a set of inliers.

The exact definition of *those which fit* is the following: two tolerances are defined, an angle tolerance ε_a on the difference of orientations (set to 5% in the experiments), and a distance tolerance ε_d on the symmetric transfer error [4] between the model and the considered match. The angle tolerance is first used as a cheap way of eliminating wrong matches, and the more expensive distance tolerance is only used if the angle fits. Note that when performing computations that involve both reference and test maps, all the distances are pre-conditioned (divided by the map size), so that they are comparable. The value of ε_d was set to 70% of the keypoint scale in the experiments, see the discussion in 3.2 for more details.

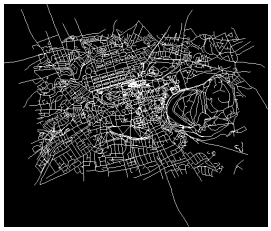
The best models (compared via the number of inliers) are stored and used in the next step. Additionally, when a computed set of inliers is larger than a preset threshold $t_{inliers}$ (20 to 30 in the experiments), the model is automatically accepted, and the random iterations aborted.

Model refinement The model refinement loop is detailed in Function RefineModels. The weights in the least-squares estimation are based on the inverse keypoint (conditioned) scales, to make inaccurate matches of large-scale keypoints less influential. This refinement step (similar ideas are in [4, 14]) increases the quality of the model estimation, and gets rid of accidentally accepted outliers.

Identification This final step uses the refined inliers and models, and identifies which references (there might be multiple) the test map originates from, assigning confidence indices to its choices (the sum of confidences is 1). The process is fairly straightforward:

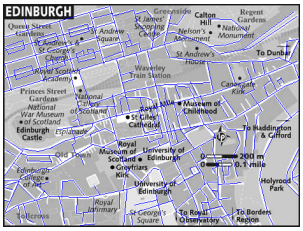


(a) Belfast reference, 14464 segments



(b) Edinburgh reference, 12604 segments

(c) Glasgow reference, 58050 segments



(d) Edinburgh test map, 306 segments. In blue, the manually selected segments.

Figure 3: Reference maps, imaged at 500px resolution (resolution in experiments is 2000px), and sample test data

1. reject reference maps with not enough inliers (10 in the experiments)
2. reject models which produce absurdly small or large test map scales ($< 200m$, or $> 10km$ in the experiments)
3. retain only the reference maps which have within $t_{f2o} = 60\%$ of the maximum number of inliers found (first-to-others threshold)
4. return the remaining reference maps, with the proportion of the total number of inliers that they hold as a confidence index

3 Experiments

This section will show and discuss experimental results and benchmarks.

3.1 Experiment data and code

All our experiments were conducted on data from the cities of Belfast, Edinburgh, and Glasgow, the metric coordinates in meters on the Ordnance Survey National Grid reference system (projected from the OSGB36 datum). The reference maps used in the experiments were all obtained from OpenStreetMap[1]. Raw XML data was then filtered using OpenJUMP[2] to retain only polylines, which were then broken down into individual segments. Figures 3a, 3b and 3c show the reference data. The test maps (e.g. Figure 3d) are tourist maps found on the internet, from which the segments were manually selected (in lieu of a road detection algorithm). They contain between 300 and 800 segments, and are of resolutions between 300 hundred pixels and 2000 pixels. The code was written in Python, using Numpy, Scipy, and OpenCV's Python interface. The source is available online[3]. Code performance was not a major concern, but for reference, reference maps were processed in 30s to 3min, test maps in about 5s, and matching performed in about 5s too.

3.2 Algorithm parameters

Here we discuss some parameters of the algorithm, their trade-offs and influence.

[1] http://www.openstreetmap.org/
[2] http://www.openjump.org/
[3] https://github.com/GautierMinster/lrpd-geolocation

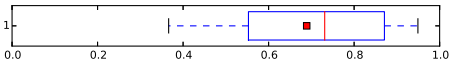
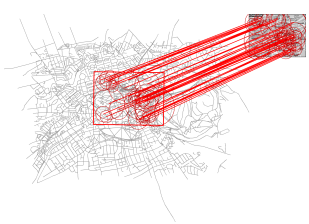
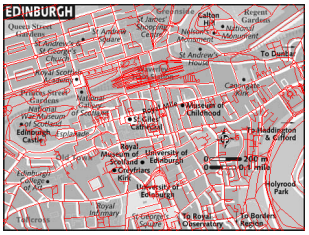


Figure 4: Ratio of smaller sum over larger sum for the halves of 19235 descriptors (all reference and test maps). 10th, 25th, 50th, 75th and 90th percentiles are shown, red dot is the mean.



(a) The 47 computed inliers. The red rectangle represents the region we estimated to be shown on the test map.



(b) OSM reference segments overlaid on the test map. The accuracy of the estimated position is about 10 meters.

Figure 5: Matching of the map in Figure 3d

Intermediate scales in octaves k: k controls how dense the scale sampling is, and has a direct impact on computational complexity and the number of descriptors. As can be expected, the number of inliers grows with k, but surprisingly, performance does not seem to degrade with high values ($k = r_{min}$ or close to it): we could indeed imagine that the very dense scale sampling would get in the way of matching by yielding too many similar descriptors, but it is not the case. A possible factor is that as the description radius changes, the whole grid of descriptors scales and shifts, so that keypoint locations are not identical. $k = 3$ was chosen, it gives good results at a relatively cheap computational cost. Note that k is not fixed: it can be chosen on a per map basis (reference or printed).

Number of histogram bins B: B can be computed from an error model for the test segments. Let X be a random variable, the angular error between test map and real orientations, and assume X follows a Gaussian distribution $N(0,\sigma)$. We want the error to be within half a bin ($= \pi/2B$), with probability p (to contribute to at least one correct bin). This yields a constraint on B, $B \leq \frac{\pi}{2\sqrt{2}\sigma\,\mathrm{erf}^{-1}(p)}$. Taking $\sigma = 5^\circ$ and $p = 0.95$ yields $B \leq 9$.

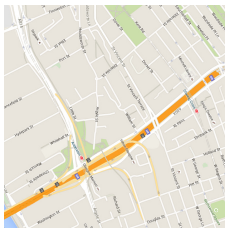
π-rotation ambiguity: the criterion we use to solve the ambiguity in 2.3.3 should clearly separate the two options, for the choice to be consistent across different maps and rotations. As can be seen on Figure 4, close to 90% of the data has a ratio lower than 0.9, which seems like a large enough margin from a ratio of 1. (i.e. total undecisiveness) to consider the criterion pertinent.

Inlier distance threshold ε_d: the RANSAC distance threshold can also be computed, by noticing that a pixel may move in a descriptor region with radius r by at most one diagonal of a histogram grid cell for it to still contribute to the same 4 histograms. The length of this diagonal is $\sqrt{2}/2 \cdot r$, so we use $\varepsilon_d = \sqrt{2}/2$.

3.3 Sample results

This section will show and describe some results.

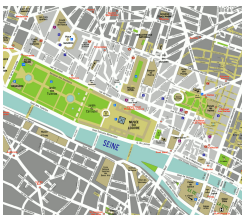
We first match as an example the test map shown in Figure 3d against our 3-cities database. 367 descriptors were computed, yielding 211 raw matches across the database (127 for Edinburgh, 31 for Belfast, 53 for Glasgow). After RANSAC model estimation, 47 inliers remain for Edinburgh, shown in Figure 5a, and none for the two other references. Using the estimated model, we can then overlay the reference data from OpenStreetMap onto the tourist map, as shown in Figure 5b. As expected given the number of inliers obtained, the algorithm correctly identifies this map as being from Edinburgh, and locates it very accurately.



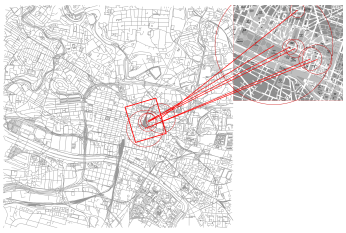
(a) A test map with unknown origin. Image ©Google.

(b) 33 inliers found with the Glasgow reference, accuracy within 1% for rotation and scaling, and around 20 meters for position.

Figure 6: Matching of an unknown map.



(a) A test map of Paris.



(b) 6 inliers found for Glasgow (and rejected)

Figure 7: Matching of a map not in any reference.

As a second example, we use the test map in Figure 6a. Ignoring the text labels, a reader unfamiliar with the geography of Glasgow will have trouble figuring out where this map is from. The algorithm produces 191 raw matches (100 for Glasgow alone), and finds a set of inliers for Glasgow (Figure 6b) exclusively. Again, the identification performs flawlessly, and locates the map (scaling, rotation and translation) with remarkable precision.

As a final example, we present the matching of a test map of Paris, i.e. a city not in the database. The test map, shown in Figure 7a, yields 599 descriptors, and 328 raw matches (116 for Edinburgh, 69 for Belfast, and 143 for Glasgow). The inlier rate is considerably lower than previously, with only 4 inliers for Belfast and Edinburgh, and 6 for Glasgow. The inliers for Glasgow are shown in Figure 7b. The identification process rejects all reference maps, since they do not meet the minimum number of inliers.

3.4 Scale and rotation invariance benchmark

To evaluate the performance of the descriptor with regards to rotation and scaling, we used 8 test maps (3 for Edinburgh, 3 for Belfast, and 2 for Glasgow), and manually determined groundtruths for their positions, rotations and scales. The maps are then subjected to arbitrary rotations in $[0; 2\pi[$ and scalings in $[10\%; 200\%]$. The rotated and scaled maps are then matched against our database, and the accuracy of scale, position and rotation estimation are computed.

Figure 8 shows the results. The position accuracy in % was computed relative to the size of the test map. For each test map, 1000 random transformations were applied. Note that the size of the transformed maps was enforced to be larger than 300 pixels in all dimensions. The results show the robustness of the proposed method against similarities.

3.5 Noise benchmark

We evaluate the effect of noise on matching by applying (separately) 4 kinds of noise to the map in Figure 3d. We model endpoint detection noise by point jitter, inaccurate maps by applying

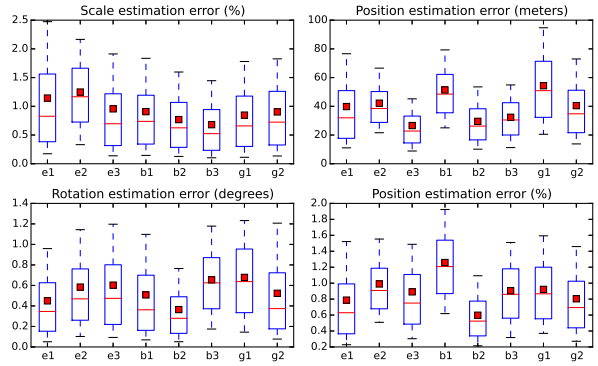


Figure 8: Accuracy of the estimation for 8 maps (1000 random similarities each, see text for details). 'e' is for Edinburgh, 'b' for Belfast, and 'g' for Glasgow. 10th 25th 50th 75th and 90th percentiles shown, red dot is mean.

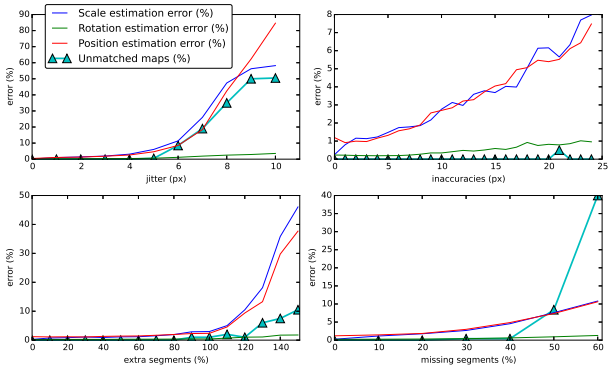


Figure 9: Median matching accuracy under different forms of noise

displacements to entire segments, and misdetections of roads by either adding extra segments, and removing existing ones.

A benchmark of the accuracy of matching against each of those 4 forms of noise is shown in Figure 9. Jitter follows a bivariate uncorrelated Gaussian distribution of mean 0 (the number of pixels is the standard deviation), and inaccuracy follows the same distribution, but applies identically to both endpoints of a segment. Extra segments are created at random positions and orientations, and have a length that follows a Gaussian of mean and standard deviation 30 pixels. The percentage shown for extra and missing segments is relative to the number of initial segments. Each data point on the plots is the median of 200 samples.

The benchmark shows the excellent resilience of the proposed method to both added segments and missing ones. Jitter is shown to be mildly inconsequent up to 5px, after which, considering the short length of many segments, performance degrades rapidly. Inaccuracies in segment positions only have a small impact on matching, which can be expected: the descriptors being poorly localized, adding to it has little effect.

4 Conclusion

We have presented an algorithm for the geolocation of printed maps, based on local descriptors of road patterns. The descriptor is heavily inspired from the SIFT descriptor [6] for natural images, and makes use of histograms of road orientations. Transposing such a texture-based matching method to the textureless context of road networks allows us to make use of a particularly well-studied matching framework, using descriptor indexing, nearest-neighbor matching and outlier robust transformation estimation.

Experiments and benchmarks show the precision of the geolocation, and its robustness to multiple forms of noise. The algorithm however suffers from poor localization of its numerous descriptors, because of the dense sampling in space and scale that was used to palliate the difficulty of keypoint and scale selection. Further work will delve into these limitations, in order to make the method scalable to immense reference regions and wider ranges of test map scales.

Acknowledgements

This work was supported in part by JSPS Grant-in-Aid for Scientific Research(S) 24220004.

References

[1] S. Callier, H. Saito, and G. Moreau. Real time detection and tracking of printed maps based on road structure. *ITE Transactions on Media Technology and Applications (MTA)*, 3(1):85–94, 2015.

[2] J. Crowley and A. C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):156–170, March 1984.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[5] D. G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 2, pages 1150–1157 vol.2, 1999.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[7] S. Martedi and H. Saito. Augmented fly-through using shared geographical data. In *International Conference on Artificial Reality and Teleexistence*, pages 47–52, 2011.

[8] A. Morrison, A. Mulloni, S. Lemmelä, A. Oulasvirta, G. Jacucci, P. Peltonen, D. Schmalstieg, and H. Regenbrecht. Mobile augmented reality: Collaborative use of mobile augmented reality with paper maps. *Computers and Graphics*, 35(4):789–799, Aug. 2011.

[9] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application (VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[10] D. Reilly, M. Rodgers, R. Argue, M. Nunes, and K. Inkpen. Marked-up maps: Combining paper maps and electronic information resources. *Personal Ubiquitous Computing*, 10(4):215–226, Mar. 2006.

[11] G. Reitmayr, E. Eade, and T. Drummond. Localisation and interaction for augmented maps. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 120–129, Washington, DC, USA, 2005. IEEE Computer Society.

[12] J. Schöning, A. Krüger, and H. J. Mller. Interaction of mobile camera devices with physical maps. In *Adjunct Proceeding of the Fourth International Conference on Pervasive Computing*, pages 121–124, 2006.

[13] D. Wagner and D. Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In *Proceedings of the 12th Computer Vision Winter Workshop*, CVWW'07, pages 139–146, 2007.

[14] L. Yang, J.-M. Normand, and G. Moreau. Robust random dot markers: Towards augmented unprepared maps with pure geographic features. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pages 45–54, 2014.

[15] L. Yang, J.-M. Normand, and G. Moreau. Augmenting off-the-shelf paper maps using intersection detection and geographical information systems. In *14th IAPR International Conference on Machine Vision Applications*, May 2015.