

Urban photograph localization using the INSTREET application—accuracy and performance analysis

Michał Grega · Seweryn Łach

Published online: 20 June 2013

© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract The paper proposes a solution to the problem of geolocation of photographs by using an algorithm to compare their content against a geolocated database of street view images, and analyzing the performance of the algorithm. The algorithm makes it possible to pinpoint the location where a photograph was taken. In order to solve this problem, we propose an algorithm based on MPEG-7 features. The paper also describes the results of optimizing the performance of the algorithm and its accuracy. We show that the algorithm scales with the size of the reference database at least up to 130 km², which was the largest urban area we tested the algorithm on.

Keywords Street view · Geotagging · Geolocalization · Urban environment · Safety

1 Introduction

Many cameras available today feature a GPS module. It makes it possible to add the geographic coordinates of the location where the photograph was taken to its metadata (EXIF). However, if such information is either unavailable or has been removed from the photograph metadata, the photograph's location may be difficult or impossible to determine.

The presented work was supported by the European Commission, in an integrated project INDECT (Grant number 218086).

M. Grega (✉) · S. Łach
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: grega@kt.agh.edu.pl

S. Łach
e-mail: sewlac@gmail.com

This problem is frequently encountered by police officers during their investigations. If the outcome of the investigation starts depending on the geolocation of the photograph, the problem becomes crucial. In this case, the only way to geolocate the photograph is to hire analysts whose job it is to manually search for the most likely location. This work is extremely labour-intensive and has a low probability of success.

The initial version of this algorithm was described in [3]. This paper outlines the principles of the INSTREET algorithm, presents the optimizations which have been added since the first version of the algorithm was published, and shows the new and extended results of accuracy tests. Our algorithm is based on low-level image features and uses MPEG-7 descriptors for the task. Data from a street view application is used as a source database of reference images. We show that our algorithm is able to pinpoint the location of the photograph based on a single characteristic detail at a cost of the high computing power required for the task. We aim at the accuracy allowing for identification of a single building, that is approx. 15 m. We also show that the algorithm scales well as the area covered by the reference database is increased.

The problem is yet to be solved by the scientific community, although it has been approached in several research papers and its importance has been stressed by the US National Intelligence Office. When compared to other solutions presented in literature, our solution provides satisfactory results tested over a major dataset. Details of the state-of-the-art analysis are presented in Section 2 of this paper.

The rest of the paper is structured as follows. Section 2 presents the architecture of the solution. Section 3 describes the results of performance optimization and accuracy tests. The paper is concluded in Section 4.

2 State of the art

The problem considered in this paper has been approached by researchers in the past. Hays and Efros [4] propose a system, similar to the one presented here, that attempts to obtain geolocation based on a single image. In their paper, the authors use geolocated images obtained from an image sharing service. Seven different features are compared in terms of accuracy of localization (where an accuracy of 200km is considered satisfactory). This concept is further extended by using the tagged reference database in [2].

Unlike the authors of [4], we focus on street view data and try to pinpoint the location of the photograph exactly. In their next paper [5], the authors focus on using a larger database and optimizing the geolocation algorithm. Work presented by Zhang and Kosecka [10] is the closest in concept to that presented here. The authors attempt to solve the problem by applying the SIFT [6] algorithm. A great effort is made to compensate for perspective transforms resulting from the different angles the photographs may have been taken from. In our work, we attempt to use the SURF algorithm [1] (enhanced and more efficient when compared to SIFT); however, it was shown to be less accurate than our MPEG-7 approach.

Schindler et al. [9] propose a solution where the location is estimated by detecting and matching repeated patterns. While the proposed solution offers a very high accuracy with a mean error of just 6 m, the paper does not propose a large scale (in terms of area covered) test of this solution. A good overview of the current

approach to geotagging research is presented in [7], where the topic is presented in terms of three different approaches: how the information can be extracted, where it can be applied, and what the interplay is between the extracted geotagging information and applications.

The importance of this topic is further confirmed by a call published by the US National Intelligence Office: Intelligence Advanced Research Projects Activity (IARPA) for a solution to the presented problem (Solicitation IARPA-BAA-11-05, May 2011).

3 Solution architecture

In order to be able to geolocate a photograph, we follow a method frequently used by specialists focusing on architectural details characteristic to specific places. We propose a complex algorithm that requires high computing power while being fully automated. The algorithm is presented in Fig. 1.

In brief, our algorithm accepts a fragment of the localized photograph as an input. This rectangular fragment is chosen by the user and should contain a characteristic architectural detail of the photograph. For instance, it could be an unusually shaped window in a building, or a piece of wall of a distinctive colour, texture and/or shape. Low level descriptors of this fragment are compared against low-level descriptors of photographs coming from a street view service.

A street view service is a popular Internet application that combines interactive maps with panoramic photographs taken at street level. Examples of such services are StreetView from Google, NORC (which covers cities in mid-eastern Europe not covered by Google StreetView) and Microsoft Streetside. Photographs from these services are of high resolution and quality, as well as being geolocated; as such they create a perfect reference database for the INSTREET application.

Our approach uses a cascade of a texture descriptor (Edge Histogram) and four colour descriptors. Afterwards, a weighted rank algorithm is used. Finally the k -nearest neighbor algorithm (k -NN) returns the candidate panoramas. The best candidate street view panoramas with their location are chosen by the algorithm and presented to the user for the final overview and geolocation of the photograph. The following sections describe the subsequent stages of the INSTREET algorithm.

3.1 Stage 1 — inputs and pre-filtering

As already mentioned, the proposed algorithm requires the user to choose a characteristic object (a.k.a. region of interest) from the photograph (Fig. 2) to be geolocated in the process. It is very important that the user marks a homogeneous set of the most distinctive objects in the photograph, for example a gateway-door-window set. This operation has a crucial role in the classification process. The tests conducted showed that if the selected object was too detailed (for example a letterbox) or too general (a whole photo), the overall quality of the result decreases significantly.

The user is also required to mark the region to be searched on the map. Our algorithm is optimized to work in an urban environment. The larger the area the user chooses, the more street view panoramas need to be searched and the longer the search time.

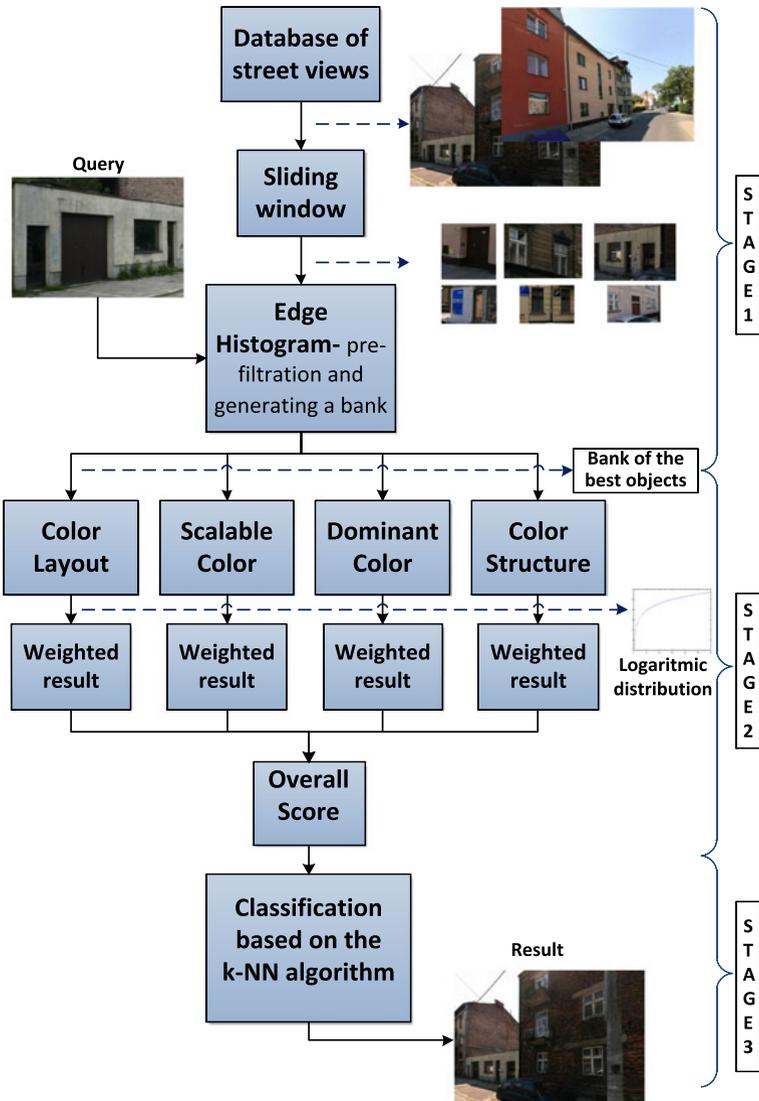


Fig. 1 Architecture of the INSTREET

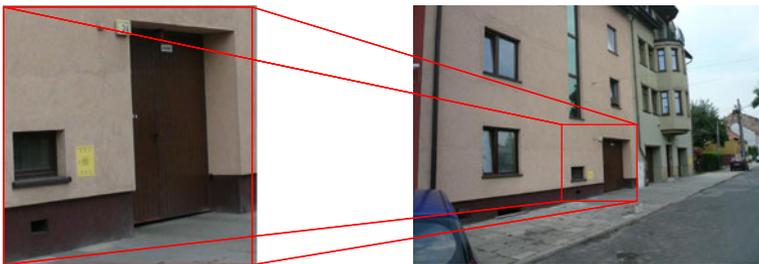


Fig. 2 Selection the region of interest

The panoramas from the selected region are divided into subimages using the sliding window algorithm. The mode of operation of this algorithm is characterized by two parameters: an area of overlap between two adjacent blocks, and a size of sliding window in consecutive iterations. In the system presented, the best results were achieved for 90 % of overlap area and 10 % of resizing applied to the sliding window per iteration. According to these parameters, one panoramic view is divided into approx. 30,000 subimages, while one street may consist of 40–80 views. This means the INSTREET is a computing-heavy algorithm.

Pre-filtering is conducted by using the edge histogram algorithm from the MPEG-7 standard [8]. The edge histogram is an algorithm that makes it possible for an image to be reduced to a vector that represents the numbers and directionality of edges in the image. Moreover, it makes it possible to compute the distance between two vectors, which in turn makes it possible to estimate the visual similarity between the two images. The edge histogram distinguishes five types of edges in local image regions, four directional (vertical, horizontal, 45°, 135°) and one non-directional.

This information is provided for every subimage. During the process of calculating the edge histogram, subimages are defined by dividing the image into 16 non-overlapping parts. As a result, the edge histogram returns a vector of values, also referred to as a descriptor value. The distance between two images is computed following the MPEG-7 standard [8]. This distance represents the visual similarity between the query and the analyzed image.

At this stage in the algorithm, the characteristic object chosen by the user is compared using the edge histogram descriptor against subimages created by the sliding window algorithm. A list of best candidate subimages is created and continuously updated during the process of pre-filtering. This list is referred to as a bank of results. Such an approach enables optimized memory management and a more time-efficient operation of the application.

3.2 Stage 2 — colour-based filters

At this stage, the INSTREET compares the candidates in the bank against the user-selected region using colour-based MPEG7 descriptors. They are the Color Layout, Scalable Color, Dominant Color and Color Structure. For each of the descriptors, all candidates in the bank are compared against the user-selected region. The results are ordered from best to worst match.

Using four different descriptors caused a problem with the final classification, which was resolved by using a weighted classification—the ranks algorithm based on a logarithmic distribution. The algorithm proceeds as follows. Each result from each descriptor is assigned a rank ranging from one to the size of the bank. The rank represents the similarity between the user-selected region and the result, where one is the most similar. Every image has four ranks assigned, one for each descriptor. The INSTREET penalizes results for being low ranked by the descriptors using the logarithmic distribution. This distribution was selected following a set of experiments. The overall score for one object is expressed by the following equation:

$$R(i) = \sum_{j=1}^4 \log[X(i, j)] \quad (1)$$

where:

- $R(i)$ overall rank for i -object
 $X(i, j)$ result for i th object in j th-descriptor (CL,SC,DC,CS)

As a result, the final output of this stage of the algorithm is a set of candidate images with an associated overall score. Summarizing, first we analyze the data focusing on the layout of the edges and later on we focus on the colour characteristics. Our experiments have shown that such approach yields best results in terms of the accuracy.

It should be noted that the presented algorithm will not be effective if the example photograph is taken at night due to colour differences between daytime and nighttime photos. In order to compensate for different light conditions, a colour normalization algorithm is currently under development.

3.3 Stage 3 — final classification

The final stage of the algorithm is to utilize the k —nearest neighbour ($k - NN$) algorithm. This is a method for classifying objects based on the closest distance in a feature space. After the logarithmic classification, a set of closest-matching photos is available. The role of the $k - NN$ algorithm is to assign objects to their own model (a street view). As a consequence it is possible to determine which class of objects is the most numerous in the $k - NN$ classification. The algorithm outputs the best candidate panoramas.

4 Experiment and results

This section describes the practical experiments performed with the INSTREET algorithm in order to assess the accuracy and time performance.

4.1 Reference database

For reference, the database in the INSTREET system uses street view images from one of the popular systems. Our reference database was organized into views, and the views into regions. One view covered a panoramic, 360° image from the street view level. A set of such views from one city district creates a region. Table 1 presents the statistics of views and regions. Columns 3 and 4 refer to the area of the municipality of Krakow, Poland, where the experiment was conducted.

Table 1 Views and regions in the reference database

Region ID	# of views	Area [%]	Area [km ²]
1	4,840	4,8	15,8
2	8,432	8,4	27,6
3	8,932	8,9	29,2
5	10,028	10,0	32,8
7	9,500	9,5	31,1

4.2 Time performance

The time performance of the INSTREET algorithm has been greatly improved compared to the performance described in [3]. The only optimization in the early version of the application was the downscaling of panoramic views, which resulted in an acceleration of the process by a factor of three. In the version presented in this paper, we made significant progress in the descriptor calculation time and in the descriptor comparison time, accelerating the process 150 times.

As for the time required for the descriptor calculation, we have created a multi-threaded version of the software used for calculating the descriptors. The process of calculating the descriptor values is independent for each of the subimages generated by the sliding window, and can be executed in a separate thread. As the process is CPU intensive, the acceleration in processing time is proportional to the number of threads used. We used an eight-thread processor for the experiment, therefore the descriptor calculation was accelerated eight times. The time required to calculate a set of descriptors for one view is 55 s for a single thread, which means it takes approx. 26 days of computing time for the entire set. By using multi-threading, we reduced the time required to calculate the descriptors for the entire dataset from 26 days to 3 days. Further improvement is possible by using a more powerful machine.

As for the time required for the search, we switched from on-demand calculation of the descriptors to pre-calculating the descriptor set for the whole reference database. In the first version of the application, the entire dataset was analyzed each time we issued a query. This is far from optimal, although it allowed us to create a proof-of-concept of the application. The time performance of the application has since been improved by pre-calculating all the descriptors and limiting the search process just to comparing the descriptors. The largest dataset we were capable of testing with the first version of the application was Region 1. A query was calculated for 36 hr. With pre-calculation, we were able to increase the performance by a factor of 150 with a query to this region taking 13 min. This also allowed us to perform tests on larger regions within reasonable time frames. The results of the accuracy tests are presented in the next section.

4.3 Accuracy

In order to assess the accuracy of the INSTREET application, we prepared a set of 30 queries that were processed by the algorithm. The queries were selected from a different street view service to the one that was used to create the reference database. This way the queries are similar to what can be expected during the real operation of the system, with photographs of urban scenery from the street view level, although taken in different conditions (lighting, weather) than those stored in the reference database.

We analyze the results of the query by observing the returned list of results. The score given to a query in terms of accuracy is the place on the list of results where the relevant result is given (we refer to it as ‘rank’). The lower the rank, the better, as in an ideal situation we expect the relevant result to appear in the first position. We also calculate the percentage of the entire dataset that needs to be analyzed by an operator by going down the list of query results in order to find the relevant answer. We also expect this value to be as low as possible. The most important question that

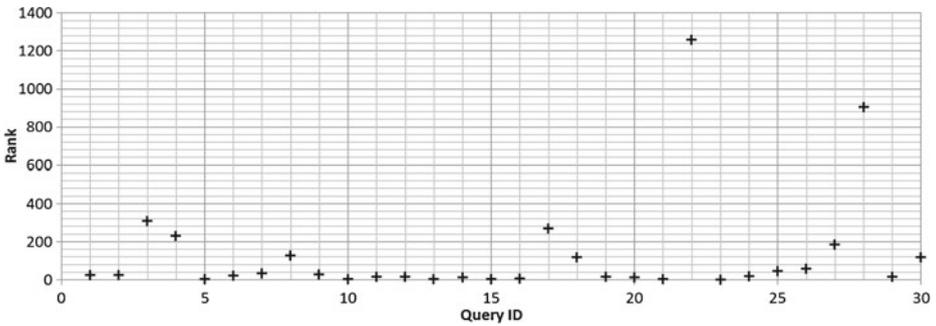


Fig. 3 Accuracy results for the query set for the whole dataset

needed answering was the impact of increasing the dataset that is available to be searched on the accuracy of the results. In other words, does the INSTREET scale as the size of the area covered is increased? In order to answer this question, we increased the size of the reference database by adding more regions in consecutive experiments. As described previously, 30 queries were issued in each experiment.

One of the problems we encountered is that in some cases the location that is photographed and submitted as a query is not available or visible in the reference database. It is impossible to create a database of photographs of every single urban setting in a city. In certain cases, the query presents a location that does not have any significant features distinguishing it from other locations. For example, residential buildings constructed from prefabricated elements, popular in mid-eastern Europe, are visually very alike. It follows that while the overall performance of the system may be satisfactory, there are some outlying queries for which the results are significantly worse. An example of such an effect is presented in Fig. 3. The vertical axis represents the rank score, while the horizontal axis represents the set of queries. In this example, the query was posed against the entire dataset including Regions 1, 2, 3, 4 and 7, covering 41 % of the area of the city of Krakow (approx. 136 [km²]). Queries 22 and 28 in the given example are evident outliers. In order to reflect this, both average and median results are given for the accuracy results.

Table 2 presents the results of the tests of the INSTREET application. The “Region” column describes the set of regions used for a given experiment set. The “# views” and “Area” columns give information on the number of views in the reference database and the area of the city covered. The “Average” and “Median” columns provide information on the average and median percentages of the result set where

Table 2 Results for accuracy tests for the INSTREET application—percentage

Region	# views	Area [km ²]	Average [%]	Median [%]	Worst [%]
2	8,432	15,8	0,22	0,04	1,80
1&2	13,272	43,4	0,26	0,06	3,07
1&2&3	22,204	72,6	0,34	0,06	2,99
1&2&3&7	31,704	103,6	0,32	0,05	3,06
1&2&3&5&7	41,732	136,4	0,32	0,05	3,01

Table 3 Results for accuracy tests for the INSTREET application—rank

Region	To analyse – average	To analyse – median	To analyse – worst
2	18	3	152
1&2	35	8	408
1&2&3	75	14	664
1&2&3&7	103	17	970
1&2&3&5&7	132	22	1,256

the relevant result was found. The “Worst” column presents the worst outlying query in the query set.

Table 3 presents additional results as a rank. It represents the results that need to be browsed by the user in order to reach the relevant one. The “To analyze—average” and “To analyze—median” columns provide information on how many results need to be browsed on average and as a median respectively. The “To analyze—worst” column presents the worst case scenario. It should be noted that while the average results may be not satisfactory, this is due to the outlying results. The median results meet our expectations, where an operator using the INSTREET algorithm would need to browse just 22 results in order to identify the location where a given photograph was taken from an area of 136 km² of an urban environment.

Figure 4 presents the relationship between the size of the dataset and the accuracy of the results. The vertical axis represents the percentage of the dataset where the relevant result can be found, and the horizontal axis the area of the city covered by the reference database. It should be noted that for all three metrics that were used, the system scales in a linear manner as the searched area is increased. In terms of the absolute numbers the user would need to browse, the more results generated, the larger the dataset. However, the median outcome is that they would only need to go through 22 results for the largest dataset.

We also created a set of experiments where the queries were user-generated. We took a set of photographs in the city and used them as queries in the INSTREET system, using the largest dataset. The first problem we encountered was that some of the areas we photographed are not indexed in the street view services used to build

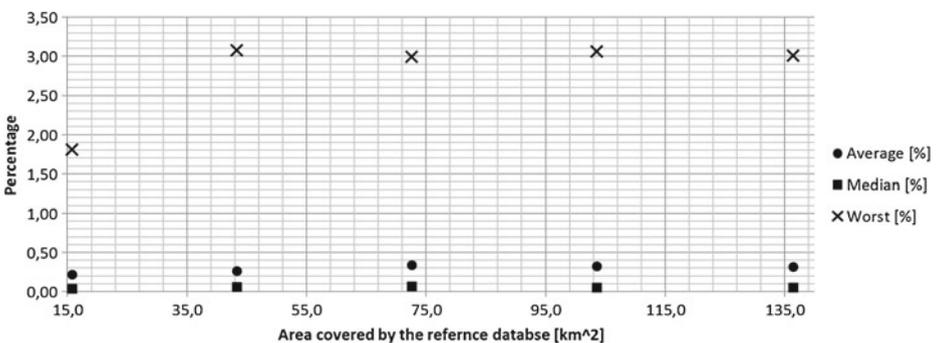
**Fig. 4** Scalability of the INSTREET algorithm



Fig. 5 Illustrative result for user generated content

our reference database. Such services are created using car-mounted camera systems, therefore pedestrian-only zones are usually not covered. Our test dataset was not large enough to perform a statistical analysis of the results, although it allowed us to confirm that the system works for user-generated content. Figure 5 shows the user-generated query and results. In this particular example, the relevant hit can be found in the fourth position of the result set ($rank = 4$)—and in the lower right hand corner in the figure.

5 Conclusions and further work

This paper proposes an algorithm for geolocating photographs in an urban environment. Our algorithm requires the user to select a region of interest in the photograph, such as a distinctive architectural detail. Based on this rectangular image, the algorithm searches a database of geolocated panoramas taken at the street view level. Using MPEG-7 descriptors, the algorithm suggests geolocated panoramas that are most likely to have been taken in the vicinity of the query photograph.

We significantly improved the time performance of the algorithm by enabling multi-threaded calculation of the descriptors and by pre-calculating a descriptor set for the whole reference database. As for the accuracy of the solution, we created a set of tests with queries generated from another, competitive street view service. The tests reveal that while a positive result is obtained for the majority of the queries, some troublesome queries remain. We also demonstrated that the system scales linearly as the urban area covered by the reference database is increased.

In further work, we plan to create a GUI for the algorithm in order to create a prototype application. We will also work on speeding up the most time-consuming operation, which is the calculation of descriptor values by delegating this task to a GPU instead of the CPU. We also plan to perform a set of experiments with user-generated content that will further demonstrate the usefulness of the algorithm. Additionally, we hope to deploy the service to a multi-threaded supercomputer and prepare a web-accessible demonstrator capable of returning search results in real time.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) SURF: speeded up robust features. *Comp Vision Image Underst (CVIU)*, 110(3):346–359
2. Gallagher A, Joshi D, Yu J, Luo J (2009) Geo-location inference from image content and user tags. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2009. *CVPR Workshops*, pp 55–62
3. Grega M, Łach S, Cyganek B (2012) INSTREET—application for urban photograph localization. In: *Multimedia communications, services and security communications in computer and information science*, vol 287, pp 130–138
4. Hays J, Efros AA (2008) IM2GPS: estimating geographic information from a single image. In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*
5. Kalogerakis E, Vesselova O, Hays J, Efros AA, Hertzmann A (2009) Image sequence geolocation with human travel priors. In: *Proceedings of the IEEE International Conference on Computer Vision Recognition (ICCV)*
6. Lowe DG (1999) Object recognition from local scale-invariant features. In: *Proceedings of the International Conference on Computer Vision 2*, pp 1150–1157
7. Luo J, Joshi D, Yu J, Gallagher A (2011) Geotagging in multimedia and computer vision—a survey. *Multimed Tools Appl* 51:187–211
8. Manjunath BS, Salembier P, Sikora T (2002) *Introduction to MPEG-7*. Wiley
9. Schindler G, Krishnamurthy P, Lublinerman R, Yanxi L, Dellaert F (2008) Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. *CVPR*, pp 1–7
10. Zhang W, Kosecka J (2006) Image based localization in urban environments. *International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT 2006*, North Carolina, Chapel Hill



Michał Grega started his university education at the University of Science and Technology in Cracow, Poland in 2001. In 2006 he received his MSc. Eng. with honors. In 2011 he defended his PhD Thesis titled “Performance Analysis of a Query by Example Image Search Method in Peer to Peer Overlays”. In 2012 he finished a post diploma study on project management at the Cracow University of Economics. Also in 2012 he finished a course on entrepreneurship and innovation at the UC Berkeley, CA, USA. In 2005 he joined a research team at the Department of Telecommunications at the University of Science and Technology, where he took part in several national and European projects. In 2008 he was appointed as a reviewer of the FP7 STREP projects on behalf of the European Commission. He is an author of almost 50 publications including 5 journal papers and

3 book chapters. His research interests include, but are not limited to the aspects of multimedia in P2P overlays, multimedia search, 2D and 3D image recognition and processing and QoE estimation for multimedia services.



Seweryn Łach started his university education at the University of Science and Technology in Cracow, Poland in 2007. In 2010 he joined to the European project- INDECT, at the Department of Telecommunications. In 2011 he presented the Bachelor Thesis “Selected road signs recognition based on analysis of statistical moments”. In 2012 he defended his Master Thesis titled “An application for localization photos in urban sceneries” He interests in programming, embedded systems, image processing and recognition.