Automatic geo-referencing of BIM in GIS environments using building footprints

Author:
Diakite, AA; Zlatanova, S

Publication details:
Computers, Environment and Urban Systems
v. 80
0198-9715 (ISSN); 1873-7587 (ISSN)

Publication Date:
2020-03-01

Publisher DOI:
https://doi.org/10.1016/j.compenvurbsys.2019.101453

Automatic Geo-referencing of BIM in GIS Environments Using Building Footprints

Abdoulaye A. Diakité*, Sisi Zlatanova

Faculty of Built Environment, GRID Lab, University of New South Wales, Sydney, Australia

Abstract

The integration of Building Information Models (BIM) in environments such as Geographic Information System (GIS) is currently one of the main investigated research problem in the urban spatial information community. Among the several issues that are involved in the adequate realisation of such type of integration, one particularly relevant to the GIS world is the geo-referencing of data coming from the BIM domain. The latter, which is inherited from the Computer Aided Design (CAD) world, often relies on local coordinate systems and thereby provides models with no or wrong information with respect to their actual geographic position. Based on well-known processes in the field of computer vision, this paper proposes an automatic framework that allows to transform the local coordinates of a BIM model into its real world geographic coordinates. Starting from a geo-referenced 2D polygon of the building (footprint or vertical projection), our method directly applies a rigid motion transformation to a corresponding 2D projection generated from the BIM model in order to obtain the transformation matrix that will align the geometry of the BIM to the map's one. In comparison with the most common approaches of shape alignment and rigid body transformation, our approach requires neither the input polygons to have the same number of points nor their sequence of points to be corresponding, allowing thereby to maximize the automation of the process while still providing accurate results. Several cases of BIM geo-referencing are presented and discussed. Furthermore, our experiments show that centimeter accurary can be reached in the alignment process.

Keywords: Geo-referencing, BIM-GIS integration, 3D urban model, Geographical environment, Digital Twins

2010 MSC: 00-01, 99-00

1. Introduction

Geo-referencing an object consist in transforming its geometric coordinates such that, when printed on a map, the object is placed at the right geographical location and is well aligned with the other map features. It is a common process in Geographic Information Systems (GIS), mainly for the integration of data from different sources and spatial analysis that account for the geographical context. This is achieved by projecting all the features on a map that uses a common coordinate reference system (CRS). Such map provides geographical coordinates based on specific projection of a global coordinate system [1]. Thereby, GIS-based urban models include geo-referencing steps by default in their processes.

On the other hand, Building Information Modelling (BIM) comes with a more construction centred point of view for modelling 3D urban features. Inherited from the Computer Aided Design (CAD) world, models from the BIM world are often represented on the basis of a local coordinate systems specific to the software tools that produce them. Consequently, they are generally delivered with

*Corresponding author

Email addresses: a.diakite@unsw.edu.au (Abdoulaye A. Diakité), s.zlatanova@unsw.edu.au (Sisi Zlatanova)

none or inaccurate information about their geographic location [2]. But since BIM models are a rich source of information at a design stage (and beyond, although they get outdated) and GIS models provide a good support for city planning, their integration became quickly a critical issue to tackle for the spatial information community [3]. Hence, the issue of geo-refencing BIM models in GIS environment became also one of the main challenge to the integration process. In theory, the most straightforward way to solve the issue would be to integrate some geodetic information related to the constructions represented by BIM models. For example, the Industry Foundation Classes (IFC) which is the most renown open BIM standard (ISO 16739-1), provides several classes that allow to describe the geographic location of the model, along with its geometric and semantic information. However, those classes are almost never informed properly in practice [2], because such information is not critical to their primary use (design phase). Therefore, the common way to solve the issue is to manually correct the localization of the BIM models with their original modelling tools, which is tedious and restrictive.

In this paper, we propose an automatic approach that solves the problem quicker and robustly. Assuming that one has, on the one hand, a map polygon feature with the correct geographic coordinates of the model to georeference and on the other hand the geometric information of the latter, our algorithm can efficiently retrieve the correct transformation matrix to apply to the geometry of the BIM model so that the latter is fully geo-referenced on the chosen map. Our method relies on a combination of several pattern recognition approaches that are utilized and improved in a way that overcome their limitations, which are requiring shapes of same number of points as input or prior knowledge of corresponding points. The 3D geometries of the BIM models are Z-projected and simplified as convex hull, similarly to the polygon features from the map, to improve the processing time while returning precise matching results. To test our approach, a set of 6 BIM models of different building styles are geo-referenced and visualized on a GIS map.

The rest of the paper is structured as follow. Section 2 explores the existing literature of methods that attempted to solve the issue. Section 3 provides step by step the details of our approach. In Section 4 we show the results of our experiments on six chosen BIM models of different complexity. Finally, we end the paper with a discussion in Section 5.

2. Related work

The need for geo-referencing BIM models has been widely recognized in the GIS literature. The problem can be seen as a feature alignment/matching issue, which has been extensively studied in the field of photogrammetry. Nevertheless, there is still a clear lack of practical solutions to address this issue in an automatic way. In this section we will explore the most relevant works dealing with the problem.

2.1. BIM/GIS integration

The integration of BIM and GIS information has gained a great attention in the last decade [4]. BIM models are strong in representing detailed building information at a construction scale. This allows to feed a wide range of applications such as indoor navigation [5, 6], thermal comfort [7], building energy modelling [8], etc. GIS models are strong for representing large scale urban environments and thereby support a wide range of applications as well, e.g. hydrology [9], urban planning [10], and many more with the development of 3D GIS [11]. With BIM providing a highly detailed point of view, and GIS a global one, it became rapidly obvious that the integration of both domains would benefit to several applications in performing better spatial analysis (e.g. for emergency response [12, 13] or building site assessment [14]).

However the task is not trivial and several issues hinder the process [2]. The common approach that is adopted in terms of BIM/GIS integration is to bring information from the BIM domain into a GIS environment [3]. Consequently, the geo-referencing issue becomes one of the main problems in such scenario. Döllner and Hagedorn [4] emphasized on the necessity of geo-referencing CAD/BIM data for integration with GIS. They proposed a mechanism at the client side of their web service framework for the user to define the geospatial position of their CAD data, under the WSG84 reference system. Although no clear details of the process were given in the paper, the approach seems to expect the users to provide the extra information required for calculating an offset vector defining the correct geographic position of the features. It is also not clear if the rotation is taken into account in that operation. The authors acknowledged that, although the method works fine for their specific scenario, it is not a general approach as no specific syntax or semantic was well defined for the geo-referencing attributes. They expected a better formalization from the next versions of the IFC standard. Frédéricque et al. [15] also faced the problem in an application related to cadastre, but they did not propose any solution for it. Dore and Murphy [16] faced the issues of different coordinate systems in the integration of BIM and photogrammetric data (such as point clouds). The authors claimed a process for importing the data with their true coordinates in their HBIM (for Historical BIM) environment, but no detail of the process was provided. Similarly, Mignard and Nicolle [17] integrated a geo-referencing tool to handle the issue at two levels of their integration workflow: a geocoding phase during the import process, and a transformation phase of the coordinates when displaying data, if necessary. But here also, no further detail on how such operations are handled or what information they require is provided. Rafiee et al. [18] proposed a solution to the issue by relying on the geo-referencing information available in IFC models, for which the latest versions propose better geo-referencing options, as expected by [4]. Nevertheless, Arroyo Ohori et al. [2] discussed the unreliability of such information in IFC files due to a persisting lack of best practice, making the assumptions in [18] still weak for a solution to the issue. The authors in [2] proposed a manual process to geo-reference BIM data at the design phase using the Revit software, which is not a process that is always possible to execute in practice.

In summary, the geo-referencing issue is well acknowledge but still no robust automatic procedures have been presented. Most available literature do not detail their geo-referencing processes (probably solved manually, as in [2]) or assume that good geo-referencing information is already available in the model, or applied to it (as in [19]). For several reasons, we believe that manual processes are not viable solutions in the long run for the BIM/GIS geo-referencing issue:

- they are not well described in the literature and are often designed for specific projects (such as in [4]);
- even when they are described (such as in [2]), they are subjective and cannot be evaluated in terms of error and precision;
- they can hardly be reproduced as they often rely on very specific tools and interfaces;
- finally, they neither encourage a systematic/standardized solution for the BIM/GIS issue (as mentioned in [2] and [4]) nor support scalability for big projects (e.g. multiple buildings integration in a GIS).

In this work we propose an automatic framework that does not assume the availability of geo-referencing information in the BIM model.

2.2. Automatic alignment methods

The problem of aligning shapes from different coordinate systems is a well known issue in pattern recognition [20], computer graphics [21] and remote sensing [22].

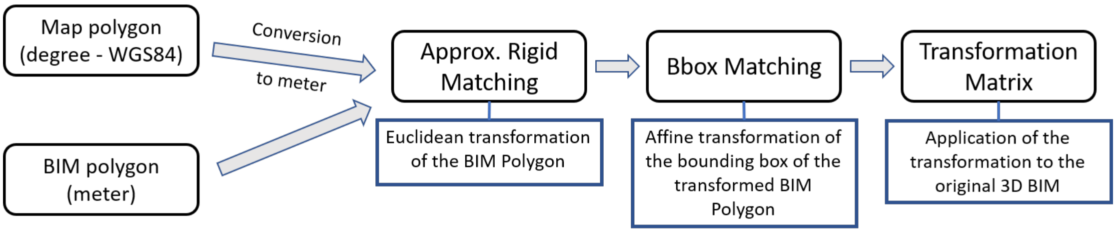


Figure 1: Work flow of the proposed method.

The situation is generally, given two shapes A and B, to measure how much they are similar and could be fit in a same space. In the latter case, one of A or B could undergo certain geometric transformations in order to be matched with the other [23]. Shapes A and B could correspond to point sets, curves, or polygons, but they would be generally represented as a sequence of points. The transformations in question are called *Euclidean* or *rigid-body* when they may involve only translations and rotations of a shape, while preserving the euclidean distances between each pair of points of the transformed shape [24]. When scaling is involved in the process, then the transformation is called affine (invariant over translation, rotation and scaling) [25].

Among the several approaches proposed to solve this problem (also referred to as the absolute orientation problem in photogrammetry), some of the most used methods are least-squares fitting [26] and the closed-form solution proposed by Horn in 1987 [27], related to robotics. The latter method is very efficient and powerful compared to its predecessors because it provides, in form a unit quaternions, the optimal solution to the least-squares problem without need of iteration or solving a system of linear equations. A minimum of 3 corresponding points from both input shapes is required to recover the 7 degrees of freedom involved in the translation (3), rotation (3) and scaling (1). However, more corresponding points provides greater accuracy in determining the transformation parameters. Umeyama later proposed a refinement of the Horn's method which would ensure a correct rotation matrix under corrupted inputs, where the other approaches would fail [28]. Several other approaches to the shape matching problem can be found in the literature, and good survey is presented in [23]. A common limitation to the previously discussed approaches is their need for prior knowledge of corresponding points in the inputs. Furthermore, the inputs should have the same number of points. This is considerably hindering the automation process, as users would be required to ensure such correspondences and size constraints before using the algorithms, which is not trivial for most applications.

In our case, a widely spread thought is that to obtain the transformation of a given BIM model with respect to its geographical location, three corresponding points should be found between them and most of the matching approaches would solve the issue. But in practice it is difficult to obtain those corresponding points because BIM and GIS data are yet to be integrated in a common environment and they often differ in dimension (3D vs 2D) and geometry. Thus, even manual accurate correspondence identification is a challenge. Currently, the common approach to the goereferencing of BIM is to manually translate, rotate and scale the models or its corresponding floor plans on the BIM platform that generated them, if the latter allows it [2]. In this paper, we introduce a method that neither require the inputs to have the same number of points nor their corresponding points to be known in advance.

3. Geometric matching of BIM and GIS maps

The process of finding the transformation matrix that aligns the geometry of a given BIM model with its corresponding feature on a GIS map is presented in Fig.1.

The two inputs correspond to a polygon on the map, which may be a vertical projection or a footprint of the building to match, and the polygon of the BIM corresponding to its Z-projection. While the input data may be different in terms of unit, scale and even shape, we include few pre-processes that allows to overcome those differences. Once they are both converted into metric shapes, the approximated matching algorithm is applied in order to get the best approximated alignment of the two shapes under euclidean transformation. From there, the oriented bounding boxes of the footprints is used to apply the final adjustments that improves the approximated matching and lead to the final transformation matrix as an output.

3.1. Map polygon

Building are commonly represented as 2D features in GIS maps based of their vertical projection or their footprint. They are generally represented as 2D polygons at specific geographic locations. Figure 2 shows the difference between a footprint data and a vertical projection. For the footprint (see Fig 2(a)), the same building is represented by three separated polygons which correspond to the parts of the building that are touching the ground. The vertical projection (Fig 2(b)) gives a more complete shape of the building. For this reason, that latter type of 2D representation gives a better input for our process, as the BIM models come as complete model and it is not a trivial process to directly identify the structures at the footprints. However, this does not mean that footprints are necessarily non-suitable. A common case is when the footprint of a building is similar to its z-projection.

The coordinates of the polygons' vertices are represented with respect to the CRS of the map. The most commonly used CRS is the World Geodetic System (WGS), for

(a)

(b)

Figure 2: Two different 2D map representation for a same building. (a) Footprints of the building. (b) Feature closer to a Z-projection of the actual building.

which the latest version published in 1984 (WGS84) is used for the Global Positioning System (GPS). The coordinates of a point are expressed in terms of latitude (North-South position) and longitude (East-West) on the Earth's surface [1]. A latitude coordinate is an angle (expressed in degrees) ranging from 0° (at the equator) to 90° (at the poles), and is positive above the equator and negative below. A longitude coordinate is an angle (expressed in degrees) between two meridians (a virtual line connecting the poles and sharing the same longitude), namely between the prime meridian (set by convention to be passing through the Royal Observatory, in Greenwich, England) and any other. Therefore the longitude at the prime meridian is 0°, spanning up to +180° towards the East and −180° towards the West. In this work, we assume that a map polygon is provided under the WGS84 CRS and has thereby the coordinates of its its vertices expressed in degrees.

The transformation that we apply to the geometry of the BIM models is affine. Therefore, we need to convert the coordinates of the map feature into a planar CRS such that it can be compared to the z-projection of the BIM. For that purpose, we use the web Mercator projection which is the most adopted projection of web map services [29]. Let $A_d(\lambda, \varphi)$ be a point for which the longitude and the latitude are expressed in degrees. The corresponding point $A_m(x, y)$ in meter would be such that:

$$x = R(\lambda - \lambda_0).\frac{\pi}{180} \tag{1}$$

$$y = R \ln \left[\tan \left(\frac{\pi}{4} + \frac{\pi\varphi}{360} \right) \right] \tag{2}$$

Where R is the equatorial radius of the Earth approximated as a sphere (rather than an ellipsoid), λ_0 is the angle at the longitude of reference. In our case, $R = 6,378,137.0m$ and $\lambda_0 = 0$ (Greenwich meridian).

3.2. BIM polygon

The geometry of a BIM model is a complex structure composed of several different objects with their own volumetric shape each. In the case of IFC models, although the standard provides a strong hierarchy of the different classes of objects that it contains, there is no straightforward way to identify which parts of the building are touching the ground and can thereby be considered to identify the footprint. Figure 3 provides an illustration of the problem. The entire model represent several floors and one underground basement level (Fig.3(a)).

By manually selecting the components of the ground floor level, we end up with what is shown in Fig.3(b). Unlike the footprints in Fig.2(a), we have two main separated components rather than three, and several features are just columns rather than slabs. Thus a projection of their shapes on the ground would still not give a satisfying result. Shapes closer to the footprint polygons, although not guaranteed, could be obtained by further manually picking the geometries to process. However this would result in a tedious process. In comparison, the top view of the full building (Fig.3(c)) gives a more satisfying similarity to the vertical projection illustrated in Fig.2(b), although there are still visible differences. Since our goal is to automate the whole process, we set our preference on map features such as Fig.2(b) in order to avoid having to manually select the BIM features to use.

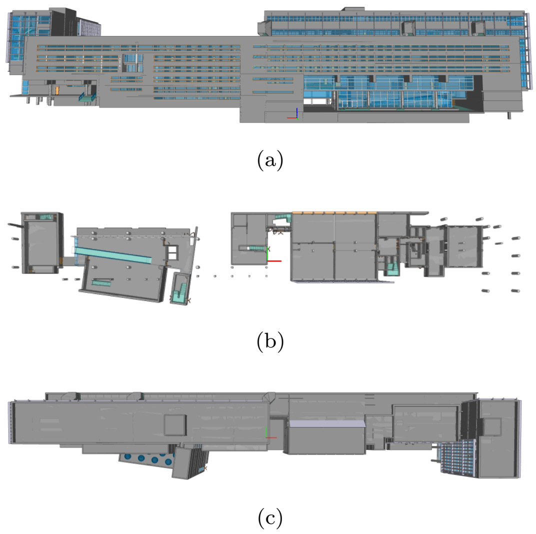


Figure 3: IFC model of a building. (a) Front view of the model. (b) Top view of the ground floor components (manually oriented to be compared to Fig.2 easier). (c) Top view of the full model.

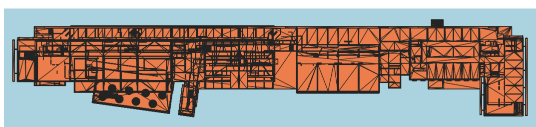
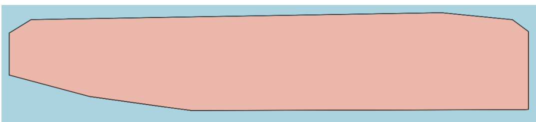
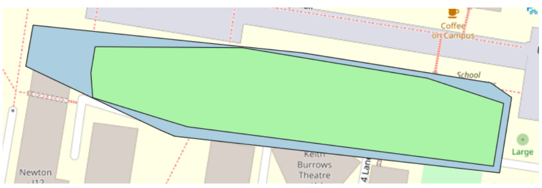


Figure 4: Flat projection of an IFC model on OpenSteetMap [30] under the Pseudo Mercator CRS (screeshot from the QGIS software [31]).

We need to obtain data from the BIM models that is comparable with the data on the map in order to find the proper transformation to apply to them. Thus, we compute the Z-projection of our BIM model on the ground (assumed to be the Cartesian XY-plane). But because a BIM is a complex 3D object, such direct projection leads to complex features for which it is not straightforward to obtain just the layout of the shell. Figure 4 shows the result of the projection on a Pseudo Mercator CRS. Note that the location of the building is not correct yet, but the complexity of its projected shape appears clearly.



(a)



(b)

Figure 5: (a) Convex hull of the projected BIM model. (b) Convex hulls of the footprints (green) and the projected (blue) map polygons.

To overcome this, we use the 2D convex hull of the projected IFC geometries, as illustrated in Figure 5(a). Similarly, we compute the convex hulls of the available 2D features from the maps assuming that similar shapes should have similar convex hulls. Figure 5(b) shows (in green) the convex hulls of the footprint polygons and (in blue) the one from the global projection of the building. The latter logically appears to be closer to the convex hull of the BIM model in terms of shape, thus both shapes are used as input to the matching approach.

3.3. The matching algorithm

Once a proper 2D feature is selected for the BIM model to geo-reference, and the coordinates of both BIM and map convex hulls are expressed in meters, the process of automatically finding the right mapping transformation can start. For this purpose, we rely on a combination of two approaches: the approximative rigid matching of Goodrich et al. ([32]) and the closed-form absolute orientation method of Horn ([27]). The first method has the advantage to provide a good approximation of the alignment of the input shapes under Euclidean transformations (line and angle preserving translations and rotations) without requiring any point correspondence to be known between the input data. Furthermore, the sequence of points of the input shapes are not constrained to have the same cardinality, which are, combined with the first advantage, a powerful lever for the automation of the process. The second method, despite having the disadvantage of requiring known corresponding points and same input size, provides in one single closed-form step the exact best matching between the shapes and can handle the scaling as well. We detail our use of those approaches and the modifications that we have brought to them to fit to our case.

3.3.1. Approximative Rigid Matching

Most of the approaches for exact or approximate point pattern matching are admittedly quite difficult to implement and sometime unstable because of some geometric operations involved in their processes [23]. Goodrich et al. provided an interesting alternative with their simple and practical method that gives good matches in a faster way, although not exact. Their algorithm uses the directed Hausdorff-distance as a metric to estimate how good the approximated transformation are. The authors described several transformation methods, but we are interested in the translation and rotation in $\mathbb{R}_2$. Let P_M and P_B be respectively the polygons from the map feature and the BIM, described as sets of m and n points in $\mathbb{R}_2$. Algorithm 1 describes the process that leads us to the Euclidean transformation matrix $T = [R, t]$ that approximates the optimal minimization of $h(T(P_B), P_M)$, where R and t are respectively the rotation and translation, and h is the Hausdorff-distance function.

Algorithm 1: Approximative Rigid Matching

Data: P_M: list of m vertices of the map polygon.

P_B: list of n vertices of the BIM polygon

Result: $[R, t]$: Rotation and translation matrices

1 **begin**
2    $(r, k) \in P_B$ such that r and k are diametrically opposed;
3   i_r = index of r in P_B;
4   i_k = index of k in P_B;
5    $hd_{min} = +\infty$;
6   **for** $(i = 1 \ to \ m)$ **do**
7      $t_{tmp} = P_M(i) - r$;
8      $P_t = P_B + t_{tmp}$;
9   **for** $(j = 1 \ to \ m)$ **do**
10   **while** $(i \neq j)$ **do**
11          $v_1 = P_M(j) - P_B(i_r)$;
12          $v_2 = P_t(i_k) - P_t(i_r)$;
13          $A =$ **getAngle**(v_1, v_2);
14          $R_{tmp} =$ **getRotation**$(A, P_M(i))$;
15          $P_{B_{tmp}} = R_{tmp} * P_t + t_{tmp}$;
16          $hd =$ **Hausdorff**$(P_M, P_{B_{tmp}})$;
17   **if** $hd < hd_{min}$ **then**
18            $R = R_{tmp}$;
19            $t = t_{tmp}$;
20            $hd_{min} = hd$;

Globally, the method operates in 3 main steps:

- identification of two vertices r and k from P_B that are the farthest from each other (lines 2-4);
- successive translation of P_B to the i-th vertex $v_i \in P_M$ such that r and v_i coincide, where $i = [1, m]$ (lines 6-8);
- iterative rotation of the translated P_B about r such that r, $v_j \in P_M$ and k are collinear, at each iteration where v_j is the j-th vertex of P_M, $j = [1, m]$ and $i \neq j$ (lines 9-15).

A value hd_{min} is maintained to track the minimum Hausdorff-distance found between P_M and the temporarily transformed P_B, and the outputs R and t are updated accordingly. As proven by Goodrich et al. in [32], $h(T(P_B), P_M)$ is guaranteed to be at most $4 \times h_{opt}$, with $h_{opt} = h(T_{opt}(P_B), P_M)$ where $T_{opt}(P_B)$ is an optimal Euclidean transformation among all the valid transformations possible. Furthermore, the algorithm can be implemented to perform in $O(n^2 m \log n)$.

Although there is no emphasis on the choice of the vertex pair $(r, k) \in P_B$ in the literature, they are critical parameters to the whole process. In the original method, they can be randomly picked as long as they are diametrically opposed to each other. But a closer look at how the algorithm works shows a clear limitation in such arbitrary choice. Indeed, the vertex r (which stands for "representative" vertex of P_B in the transformations) is the point that is translated and is also the center of rotation of P_B. This means that depending on how it is picked, hd_{min} is likely to be closer to its upper bound ($4 \times h_{opt}$) rather than its lower bound (h_{opt}). This can be partly explained by the fact that the Hausdorff distance is mostly suitable for point sets comparison rather than curves or areas [23]. In fact, an area-related metric would be more suitable for our case, however this would involve expensive and precision-sensitive operations (boolean operations, boundary intersections, etc.).

Since our main goal in this step is to obtain an initial alignment that removes the rotation ambiguity between P_M and P_B, we therefore introduced an improvement on the choice of r by selecting the vertex that is the farthest to the centroid of P_B. This empirical choice is guided by the hypothesis that if the shapes of P_B and P_M are similar enough, the farthest points to their respective centroids should be located at a nearly similar position on their boundaries. This provides a good approximation of

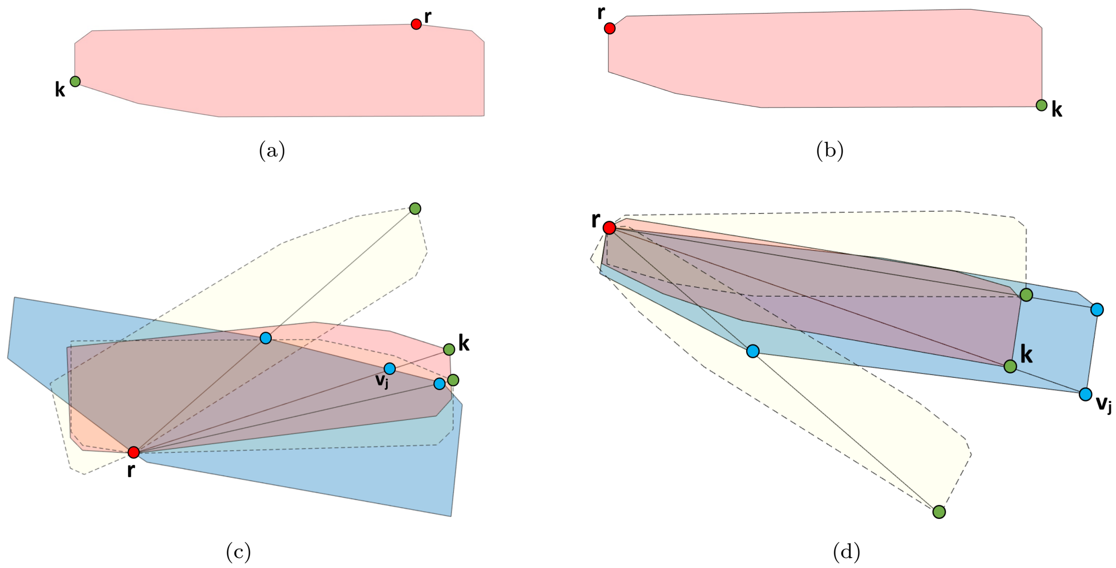


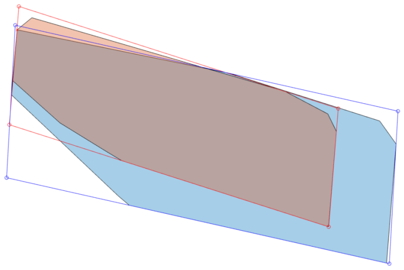
Figure 6: Different choices of r (red point) and k +(green point) leading to different matching results. The blue points correspond to v_j on the j-th rotation. (a) Randomly chosen r and its corresponding k on P_B. (b) r chosen as the farthest point to the centroid of P_B. (c) Result for (a); three selected rotations of P_B around r (in light yellow, dashed boundaries), including the optimal one found (light red polygon). (d) Result for (b).

corresponding corner points between the two input data, which constraints the Hausdorff-distance to be minimal when the shapes overlap the most. Figure 6 illustrates a case where random choice of r is compared to our approach. With the random choice (Fig.6(a) and (c)), the resulting hd_{min} coincides with a position of P_B that is angularly inverted in comparison with P_M. This is a good illustration that the Hausdorff distance is not concerned about the outline of the shapes, but when the rotation is guided by our choice of r, the result is closer to our expectation (Fig.6(b) and (d)).

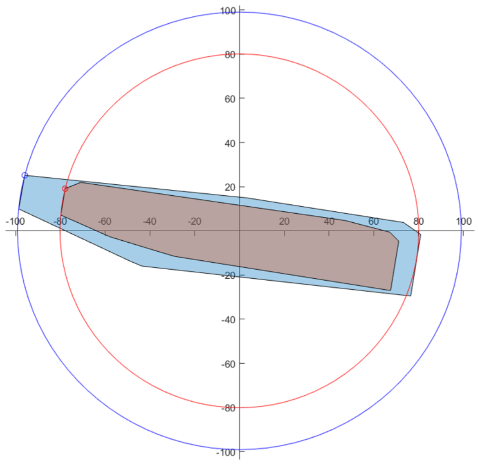
However, some situations can still challenge our improvement on the choice of (r, k). For example, when several points of the boundary of P_B happen to be equally the farthest points to its centroid (e.g. on buildings with round shapes), then an ambiguity appears with respect to the correct corresponding point in P_M. Similar issue occurs when there are significant differences in the convex hulls of P_B and P_M (e.g. a significant protrusion on one shape is missing on the other). In both cases, a solution to those limitations can be the introduction of a more advanced point similarity detection step between the two shapes, with additional constraints (e.g. angular comparison of the ambiguous corners).

3.3.2. Bounding Box matching

At this stage of the process, we have our shapes P_M and P_B approximately aligned. Although the translation is satisfactory, the rotation and the scale are yet to be determined more precisely to provide a more accurate matching. Figure 7 shows the output of the approximative matching and the difference of rotation and scale between the shapes. In Fig.7(a), it appears that our approximation of the translation is good, and even the rotation difference is slight, but P_M appears bigger than P_B. This is even more visible in Fig.7(b) where centroids of both shapes are moved to the origin and bounding circles are displayed. The figure suggests that the scale distortion is homogeneous in all directions, as concluded by the experiments of Uggla and Horemuz who explored the distortion effects of georeferencing BIM data [33]. They tested three different methods and our approach fits in the same category as their *method 1* which consists in projecting the engineering coordinate system of the BIM into a cartographic coordinates. (see chapters 3 and 4 in [33]).



(a)



(b)

Figure 7: Difference in rotation and scale. (a) Output of the approximative matching and the bounding boxes of the shapes. (b) Scale comparison of the centred shapes through their bounding circles.

With this new configuration, we have all the information necessary to perform an absolute orientation process between P_M and P_B. Indeed, as expressed earlier in this paper, the main reasons why we could not use the absolute orientation technique from the beginning is that we did not meet the input conditions required, which are input point sets of the same cardinality and ordered following their correspondences. Thanks to the approximative matching, we now have a good approximation of what the optimal transformation should look like. Let T_1 be the transformation resulting from the approximative matching. By relying on the bounding boxes of P_M and $T_1(P_B)$, we have exactly four points from each shape that we know their correspondences. This is more than the minimum of three points required by the absolute orientation method of Horn [27], we therefore use it to determine T_2 which complete our matching operation.

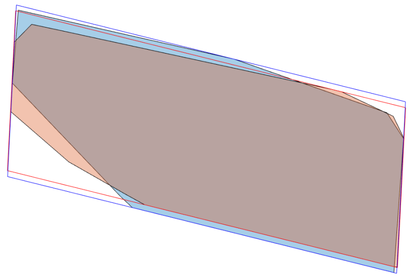


Figure 8: Best matching result of the bounding boxes with the absolute orientation approach.

Figure 8 shows the resulting final match. It can be observed that $T_2[T_1(P_B)]$ has now a comparable scale to P_M, and although their shapes are not perfectly aligned, they are aligned to a satisfactory level. This result corresponds to the transformation that minimizes the sum of squares of the residual errors between the matched shapes, and is not expected to be a perfect match of all the corresponding points, as clarified by Horn [27]. Furthermore, a perfect match would not be possible because of the differences of shape between P_M and P_B. Thereby, with T_1 and T_2 in hand, we define:

$$T = T_2 T_1 \tag{3}$$

as the final transformation matrix that should be applied to P_B such that:

$$P'_B = T P_B \tag{4}$$

where P'_B is the IFC polygon that is now georeferenced. Note that during all the process, we have been working in $\mathbb{R}_2$. However, the vertices of the model that we are aiming to transform are in $\mathbb{R}_3$. While their x and y coordinates corresponds respectively to their longitude and latitude in our process, their z components is assumed to be their correct height value (in meter), and are therefore not altered. Thus, T is expressed in the form of a 4×4 homogeneous matrix and T(row 3, col 4) is set to 1.

4. Experiments

4.1. Dataset and Implementation

To validate our approach, we implemented the method and tested it on several BIM models. We were provided with 6 IFC models of multi-storey buildings of different shapes and complexity from University X by their Estate Management (EM) department:

- *RC*, a detailed building model (51MB) describing a two-wings structure with a complex vertical shape;
- *BH*, a highly detailed model (122MB) with a relatively simple projection layout;
- *Dt*, a façade-focused lightweight model (1.78MB) with a simple rectangular vertical projection;
- *Lb*, a façade-focused lightweight model describing the aggregation of several buildings forming a single block (9.19MB) with a complex projection layout;
- *Qd*, a façade-focused lightweight model (10.5MB) with a complex layout including a combination of round and quad shapes;
- *RH*, a highly detailed model (135MB) with a mostly round and complex shape.

We relied on several open-source tools to reach our goal. Table 1 lists the selected building models and provides the number of points of their map convex hull (which correspond to the m parameter for P_M in Algo. 1), the number of points of their IFC convex hull (n parameter for P_B) and the computing time of our matching algorithm. The time was estimated on the basis of a Matlab implementation performed on an Intel Core i7 -7600U 2.80 GHz CPU laptop, with 16Gb of RAM. The IFC++ [34] library was used to extract the geometric data from the IFC files and the CGAL [35] library was used to perform operations on the extracted geometries (simplification, convex hull and 2D projections). The map layouts/footprints were either extracted from OSM [30] or provided by the EM department; in both cases their convex hulls and geographic coordinates were obtained using QGIS [31]. Note that apart from Matlab, all the tools (IFC++, CGAL, OSM and QGIS) are free open source libraries and software.

Model	Map CH (#points)	BIM CH (#points)	Time (s)	Error max (m)
RC	9 (OSM)	14	0.081	1.2444
BH	8 (OSM)	9	0.089	0.8640
Dt	9 (EM)	12	0.084	0.0616
Lb	17 (EM)	10	0.150	6.3894
Qd	59 (EM)	89	1.552	0.7814
RH	45 (EM)	120	1.020	0.2922

Table 1: IFC models used to test the geo-referencing method. The *Model* column gives the name of the buildings, *Map CH* and *BIM CH* provide the number of points describing the convex hulls of the map feature and the projected BIM (the origin of the map features is mentioned as well), *Time* indicates the time (in seconds) for the algorithm to find the optimal transformation and *Error max* provide the maximum error distance (in meters) between the aligned shapes.

4.2. Results

Figure 9 illustrates the results and depicts the major steps of the process. The left-most column shows the BIM models that need to be geo-referenced. We have several types of buildings, with different layouts and complexity. For example, *BH* (2nd row) and *Dt* buildings (3rd row) have rather simple layouts. This could be a limitation for the matching as it would be the case for basic shapes like rectangles, square or circle (a perfect match would not necessarily ensure a correct orientation of the shapes). However, small differentiations on the boundaries allow

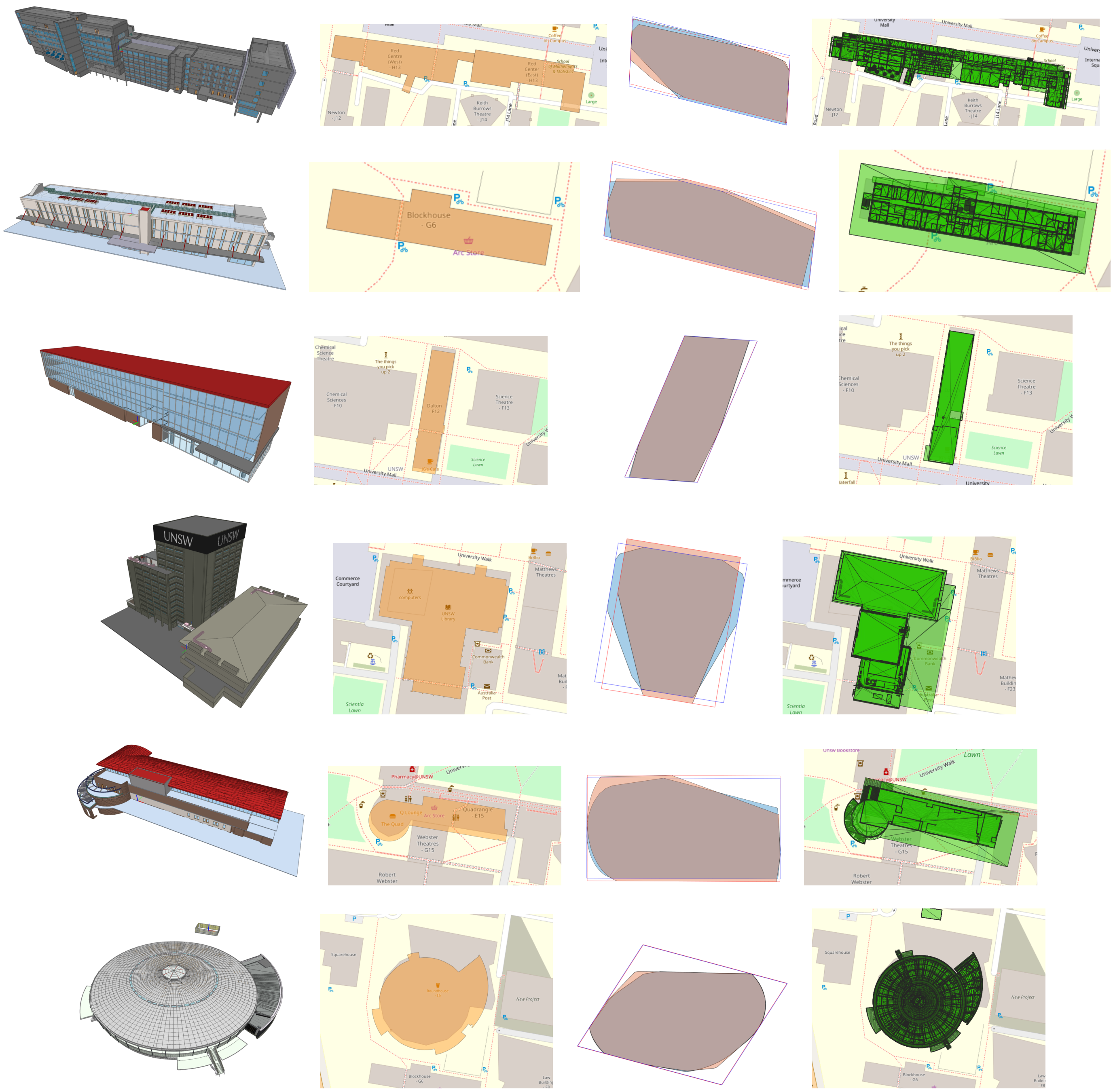


Figure 9: Results of the geo-referencing. The first column (left-most) shows the original IFC models (not geo-referenced and corresponding respectively to the rows in Table 1); the second column corresponds to their map polygon (footprint or projected layout); the third column shows the best match obtained by our method; the last column (right-most) shows the final result of the geo-referenced geometries of the IFC models on the map.

to get enough information to find the proper orientation, even with the convex hull simplification. All 2D features from EM correspond to footprints of the buildings while those from OSM correspond to vertical projections. As explained in Section 3.1, although vertical projections work better for our approach, for several buildings, footprints and vertical projections happen to produce similar map shape. Therefore, footprints are suitable in those situations as well. However, because the data from EM were obtained by precise land surveying work, they guarantee a better precision in contrast to the OSM data that were produced by a user on the basis of aerial imagery, with no guarantee of precision. Thus, we use the EM data in priority whenever it corresponds to the vertical projection of the BIM to be geo-referenced. Manual removal of some IFC features (not present in the footprint) may be required from the BIM model before computation of its polygon, in order to fit the footprints. It is the case for example for *Lb* building (4th row) for which one slab is biasing the footprint layout.

Figure 10: 3D visualization of the geo-referenced Red Centre model in QGIS.

In general, the number of points of the map and BIM polygons are relatively small thanks to the convex hulls simplification (see Table 1). Therefore the computation time is small accordingly. For some complex models such as *Qd* and *RH*, the number of points is higher due to their round boundary parts. These latter become tessellated into several smaller segments that approximate their curves and may belong to the convex parts of the whole shape. The number of points of the map polygon is more critical to the processing time however, since the steps of Algo.1 depend mainly on it.

4.3. Precision and Limitations

The right-most column of Table 1 lists the maximum error (in meter) involved in the process, which gives an idea of the precision of the approach. It corresponds to the maximum distance between two matched points after the absolute orientation process, which is ultimately a single step least square minimization process. In fact, errors are involved in both matching approaches that we combine (see Section 3). However, our approach is designed such that the error related to the approximative rigid matching algorithm does not add-up to the one of the absolute orientation on the bounding boxes. Therefore, the error from the latter process is the best measurement to reflect the error in the final alignment.

Naturally, the closest the input shapes are, the smallest the error will be. For example, the smallest errors are obtained with *Dt* (0.06m) and *RH* (0.29m) buildings, because as it can be seen in Fig.8, their footprints (provided by the EM, hence with a good precision) and their BIM models are very similar from a top point of view. This is not the case for *Lb* (6.39m) which offers the worst error due to considerable differences in its projected shape, regardless of the outdoor slab which is modelled in the BIM but not in the footprint (such feature is not considered in the convex hull computation). *RC* also presents a non-negligible error (1.24m) which is related to its approximated footprint from OSM.

Figure 10 shows the geo-referenced RC model in its geographical context. The 3D geometry of the model was imported in QGIS for a complete immersion in a GIS environment. The 3D visualization features of QGIS being still under development, thus limited, we coloured the whole model in gold yellow for visibility purpose.

4.4. Discussions

Globally, the experiments demonstrate a robust method which can offer automatic geo-referencing of BIM on a GIS map with centimetre to sub-meter to centimetre precision, depending on the input data's accuracy (BIM and map feature). However, several particular situations not covered in the experiments require further discussions.

One of them is the ambiguous case of buildings with perfectly symmetrical shapes (U-shape ,H-shape, perfect square or rectangle, etc.). As mentioned in Section 4.2, in such case our shape matching approach would provide a highly accurate alignment but would not be able to guarantee the correctness of the final orientation, due to the similarity of the convex-hull of such shapes in all orientation. As a consequence, important building features (e.g. front door) may be wrongly placed. Relying on the original polygons rather than their convex hulls for the matching may only solve the case of the U-shaped buildings however. Thus, mitigating those shape ambiguities would require additional information represented in both the map feature and the BIM (e.g. a garage nearby the building, known positions of main doors, etc.), otherwise a manual intervention to rotate the building will be necessary. Our experiments have shown, however, that any irregularity on a building's vertically projected shape (e.g. balcony, entrance cover, etc.) can be enough to overcome this issue (see *BH* and *Dt*), and we argue that such features are very likely to be present on buildings' façade.

Other interesting cases include when a BIM model is slightly or significantly different from the building represented on the map. This implies differences in the input shapes, accordingly. While our full approach is not designed to handle those cases, part of it (the approximate matching phase) can still handle partial similarities in theory, but at the condition that the scales of the shapes are the same. This is unfortunately not the case considering that the map features embed some distortions, as discussed in Section 3.3.2. Thereby, a prior knowledge of the scale factor between the shapes could improve the automation and enable the geo-referencing of new or renovated designs.

Furthermore, the fact that our approach is designed to solve the geo-referencing of single BIM models rather than several buildings simultaneously may question its scalability. However, we argue that thanks to the automatic approach that we propose, we open the possibility to automatically reproduce the process on several buildings by a simple iteration over their pairs of required inputs.

5. Conclusion and Future Work

In this paper we have introduced a work flow that allows to automate the geo-referencing process of BIM models in GIS environment. Starting from the original 3D shape of the BIM and a 2D map feature of the corresponding building, we generate a 2D projection of the 3D model and successively apply one Euclidean and one affine transformation to its shape to find the transformation matrix that best aligns both 2D shapes. The obtained transformation matrix is then applied to the original 3D model and visualized in a GIS environment. Our algorithm is efficient, robust and easily implementable with open source or proprietary spatial information software products. We successfully tested it on several real world data and, in contrast to the available literature, showed that it is possible to automate the process. Our contribution can help in bridging the gap between the BIM and GIS world to offer more integrated models for digital twin-related applications (e.g. urban planning, indoor-outdoor navigation, emergency response, etc.).

Several aspects of our approach can be improved in future work. One is related to 2D map features, which availability can be assumed only for existing buildings. Thus, further investigations and experiments need to be carried out for cases where the building is still at design phase. Furthermore, BIM models are not meant for GIS environments, therefore further studies of their integration in terms of data model is also an important challenge to address. Finally, while our approach allows mainly the geometric alignment of a BIM model in its geographic context, it mostly addresses the visual challenges of the BIM-GIS integration problem. A deeper spatial integration (semantically and topologically) of the newly aligned BIM features with the map features can allow more advanced analysis and information enrichment of both BIM and GIS models.

References

[1] J. P. Snyder, Map projections–A working manual, Vol. 1395, US Government Printing Office, 1987.

[2] K. Arroyo Ohori, A. Diakité, T. Krijnen, H. Ledoux, J. Stoter, Processing bim and gis models in practice: experiences and recommendations from a geobim project in the netherlands, ISPRS International Journal of Geo-Information 7 (8) (2018) 311.

[3] X. Liu, X. Wang, G. Wright, J. Cheng, X. Li, R. Liu, A state-of-the-art review on the integration of building information modeling (bim) and geographic information system (gis), ISPRS International Journal of Geo-Information 6 (2) (2017) 53.

[4] J. Döllner, B. Hagedorn, Integrating urban gis, cad, and bim data by servicebased virtual 3d city models, Urban and regional data management-annual (2007) 157–160.

[5] U. Isikdag, S. Zlatanova, J. Underwood, A bim-oriented model for supporting indoor navigation requirements, Computers, Environment and Urban Systems 41 (2013) 112–123.

[6] A. A. Diakité, S. Zlatanova, Spatial subdivision of complex indoor environments for 3d indoor navigation, International Journal of Geographical Information Science 32 (2) (2018) 213–235.

[7] C. Van Treeck, P. Wenisch, A. Borrmann, M. Pfaffinger, M. Egger, O. Wenisch, E. Rank, Towards interactive indoor thermal comfort simulation.

[8] V. Bazjanac, Ifc bim-based methodology for semi-automated building energy performance simulation, Tech. rep., Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (US) (2008).

[9] B. A. DeVantier, A. D. Feldman, Review of gis applications in hydrologic modeling, Journal of Water Resources Planning and Management 119 (2) (1993) 246–261.

[10] A. G.-O. Yeh, Urban planning and gis, Geographical information systems 2 (877-888) (1999) 1.

[11] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, A. Çöltekin, Applications of 3d city models: State of the art review, ISPRS International Journal of Geo-Information 4 (4) (2015) 2842–2889.

[12] H. Tashakkori, A. Rajabifard, M. Kalantari, A new 3d indoor/outdoor spatial model for indoor emergency response facilitation, Building and Environment 89 (2015) 170–182.

[13] P. Boguslawski, L. Mahdjoubi, V. Zverovich, F. Fadli, H. Barki, Bim-gis modelling in support of emergency response applications, Building Information Modelling (BIM) in Design, Construction and Operations 149 (2015) 381.

[14] J. Li, N. Li, K. Afsari, J. Peng, Z. Wu, H. Cui, Integration of building information modeling and web service application programming interface for assessing building surroundings in early design stages, Building and Environment 153 (2019) 91–100.

[15] B. Frédéricque, K. Raymond, K. Van Prooijen, 3d gis as applied to cadastre–a benchmark of todays capabilities, in: FIG Working Week, 2011.

[16] C. Dore, M. Murphy, Integration of historic building information modeling (hbim) and 3d gis for recording and managing cultural heritage sites, in: 2012 18th International Conference on Virtual Systems and Multimedia, IEEE, 2012, pp. 369–376.

[17] C. Mignard, C. Nicolle, Merging bim and gis using ontologies application to urban facility management in active3d, Computers in Industry 65 (9) (2014) 1276–1290.

[18] A. Rafiee, E. Dias, S. Fruijtier, H. Scholten, From bim to geo-analysis: view coverage and shadow analysis by bim/gis integration, Procedia Environmental Sciences 22 (2014) 397–402.

[19] L. Barazzetti, F. Banfi, Bim and gis: when parametric modeling meets geospatial data, in: ISPRS Workshop on Geospatial Solutions for Structural Design, Construction and Maintenance in Training Civil Engineers and Architects, Geospace 2017, Vol. 4, 2017, pp. 1–8.

[20] J. S. Marques, A. J. Abrantes, Shape alignmentoptimal initial point and pose estimation, Pattern Recognition Letters 18 (1) (1997) 49–53.

[21] O. Van Kaick, H. Zhang, G. Hamarneh, D. Cohen-Or, A survey on shape correspondence, in: Computer Graphics Forum, Vol. 30, Wiley Online Library, 2011, pp. 1681–1707.

[22] C. Toth, G. Jóźków, Remote sensing platforms and sensors: A survey, ISPRS Journal of Photogrammetry and Remote Sensing 115 (2016) 22–36.

[23] H. Alt, L. J. Guibas, Discrete geometric shapes: Matching, interpolation, and approximation, in: Handbook of computational geometry, Elsevier, 2000, pp. 121–153.

[24] O. Bottema, B. Roth, Theoretical kinematics, Vol. 24, Courier Corporation, 1990.

[25] M. Berger, Geometry (1987).

[26] K. S. Arun, T. S. Huang, S. D. Blostein, Least-squares fitting of two 3-d point sets, IEEE Transactions on pattern analysis and machine intelligence (5) (1987) 698–700.

[27] B. K. Horn, Closed-form solution of absolute orientation using unit quaternions, Josa a 4 (4) (1987) 629–642.

[28] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, IEEE Transactions on Pattern Analysis & Machine Intelligence (4) (1991) 376–380.

[29] S. E. Battersby, M. P. Finn, E. L. Usery, K. H. Yamamoto, Implications of web mercator and its use in online mapping, Cartographica: The International Journal for Geographic Information and Geovisualization 49 (2) (2014) 85–101.

[30] OpenStreetMap contributors, Planet dump retrieved from https://planet.osm.org, `https://www.openstreetmap.org` (retrieved on 12/04/2019) (2017).

[31] QGIS Development Team, QGIS Geographic Information System. Open Source Geospatial Foundation Project, `http://qgis.osgeo.org/` (retrieved on 12/04/2019) (2019).

[32] M. T. Goodrich, J. S. Mitchell, M. W. Orletsky, Practical methods for approximate geometric pattern matching under rigid motions:(preliminary version), in: Proceedings of the tenth annual symposium on Computational geometry, ACM, 1994, pp. 103–112.

[33] G. Uggla, M. Horemuz, Geographic capabilities and limitations of industry foundation classes, Automation in Construction 96 (2018) 554–566.

[34] Fabian Gerold, IFC++. Open Source IFC implementation for C++, `http://ifcquery.com/` (retrieved on 12/04/2019 (2019).

[35] The CGAL Project, CGAL User and Reference Manual, 4.14 Edition, CGAL Editorial Board, 2019.
URL `https://doc.cgal.org/4.14/Manual/packages.html`