3D CITY REGISTRATION AND ENRICHMENT

J. A. Quinn, P. D. Smart and C. B. Jones

School of Computer Science,
Cardiff University
{j.a.quinn, p.smart, c.b.jones}@cs.cf.ac.uk

KEY WORDS: City models, registration, shape-matching, web-mining

Geographic gazetteers are now commonly utilised to enrich digital photographs, and to make these photographs easier to locate within vast online collections. However, such gazetteers use two-dimensional data, and thus enrichment based upon complex building geometry, taking into account the high levels of occlusion within a city, is not possible. Modern cameras often provide accurate camera geometry and location, and thus 3D models are being popularised to achieve robust and accurate photograph enrichment. However, the semantic attribution within existing models is either poor or non-existent. In this work, we describe a novel and robust approach for the combination and enrichment of existing city models. We first describe the automated registration process of detailed, hand generated, 3D models to accurate 2D city ground plans, and the geometric enrichment of the output. We then describe a novel mapping function that adds existing knowledge of name and function to each building in the model using freely available web 2.0 information sources, and demonstrate the accuracy of our method.

1 INTRODUCTION

Online digital photograph collections are now widely available, and the retrieval of specific images, or sets of images, that focus on a particular subject within such collections remains an ongoing research topic (Edwardes and Purves, 2008, Purves et al., 2008). Popular existing search engines are not capable of searching image content, hence searches rely on key-word retrieval techniques from image titles and captions. However, the content of user-contributed photograph titles and captions are not usually suitable for robust geographic retrieval. Titles and captions tend to be ambiguous or colloquial and suffer from an unstructured and inconsistent vocabulary.

Recent developments in consumer level photographic devices allow positional information to be stored alongside the photograph, typically consisting of GPS positional data and camera geometry, such as pitch, roll and yaw. Information about the location can be retrieved from Web 2.0 sources, allowing the use of such geographical data sources to generate a description of the surrounding area and context. For urban environments, very rich models are appearing which can provide detailed information about the subject and geographic context of the photograph.

In practice, the level of detail of geo-data on the web is often insufficiently detailed for the scale of object that is commonly photographed within a city. In addition, where data at this scale is available, it is commonly based on 2D rather than 3D datasets. This lack of 3D geometry greatly reduces the possible accuracy of information retrieval in a dense city environment. In particular the absence of fine-grain geometric and descriptive data, in combination with what may be inappropriate assumptions about the nature of occlusion, may result in a high degree of error with respect to the actual image content when constructing photograph captions from 2D datasets.

To generate captions with high quality geographical contextual information it is desirable therefore to generate geometrically detailed and well annotated 3D city models. This process is however subject to a number of limiting factors:

- Many available, detailed, 3D building models are hand designed. In practice, whilst often highly detailed, such models are not always robustly created (e.g. geometric and topological inconsistencies, non-affine transformations, and unknowns due to a lack of design intent knowledge).
- City models may exist from a wide variety of providers, in different formats, co-ordinate spaces and at varying levels of detail and accuracy.
- Models extruded from satellite or ground plans lack fine-grain detail and building facet information.
- Currently, some grounded city models exist, for example, those on Google Earth. However, the current accuracy of such models, their placement, and level of detail can vary greatly. In addition, the hand registration of such models is very time consuming.
- Some popular modelling languages used to describe such city models i.e. KML, do not include any intricate semantic attribution about buildings and their parts, and thus building models come with little or no attribution.

In order to produce a fully annotated, accurate, 3D city model, in Section 2 we describe an approach for the combination - through robust shape matching, registration, and transformation methods - of accurate city planning data, detailed hand-designed building models, satellite imagery, and a digital elevation model (DEM). In this work, we use datasets from the city of Bamberg. Further to this, we describe web-mining methods in Section 3 for the retrieval and attribution of building data, with the aim of producing a highly detailed, accurate, and annotated 3D city model that can be used for high-fidelity photograph content retrieval.

2 MODEL REGISTRATION

In this section, we describe our approach for generating an enriched city model, by combining multiple commonly available datasets. These sources are: A 2D city ground plan, a set of high-quality triangulated 3D models of various cultural or significant parts of a city, a DEM, and satellite imagery of the city.

2.1 Approach

The 2D city data-set, P, is a set of M buildings, where each building is represented as a single, planar, polygon p_i, and $P : \{p_1, \cdots, p_M\} \subset \mathbb{R}^2$. The dataset P in this work is typical for that available from a city or council for planning applications

development, and is assumed to be the most accurate representation of the ground plans of the buildings within the city. The original data also contains roads, described piece-wise, as a set of adjacent polygons, but is not included in P (although can be re-introduced after the registration process). Each polygon p_i is also associated with an address. The 3D data-set consists of L arbitrary groups of buildings from within the city, referred to as *scenes*, $S_1 \cdots S_L$. Each scene, S_i, is assumed to be modelled as a set of N polygons $S_i : \{s_1, \cdots, s_N\} \subset \mathbb{R}^3$, which may be either connected or disjoint, i.e. no assumption is made regarding which polygons belong to which buildings, or any internal segmentation within a building. The quality of the geometry of S_i is also not assumed to be good; the topology of the scene is entirely arbitrary, holes may exist, polygons s_i may intersect, be incorrectly aligned, etc. Each polygon may be associated with one or more textures, which, if no parameterisation is supplied, is assumed to be uniformly parameterised. Each scene is therefore treated as the sort of data typically available from user-contributed services such as Google 3D Warehouse[1], being generally created by hand and not assumed to be created by a professional designer.

Due to the inaccuracy and limited coverage of the 3D scenes, S_i, and their ungrounded co-ordinate system, we wish to automatically register them to the accurate 2D data-set P. In order to register the 3D data to the 2D data, we wish to construct an injective mapping $f : S \rightarrow P$. The process of computing the mapping f involves two main steps: 1) Shape matching to find the best match in the 2D data for each 3D scene S_i, 2) Registration to find the correct transformation T for the 3D scene S_i onto it's image in the 2D data P. Once computed, the transformation is then applied to each scene, $T(S_i)$. This mapping transforms each S_i into its correct position relative to the 2D data-set P. The output of this registration is an enriched city model C, combining the available 3D data and the remainder of the 2D data.

In an effort to improve robustness, due to the unreliable quality and topology of each S_i, we compute this mapping f by registering 2D, rasterised, versions of P and S_i. Whilst varying between architectural styles, the 2D footprint of a single building is generally rectilinear, and often rectangular. This sort of shape is not ideal for shape matching within the large sets of buildings present in a city. Thus, in order to improve the chances of a correct match, we compute a binary rasterisation of S_i, and select the largest (based on area) contiguous cluster of buildings $G_i \subset S_i$, resulting in a far more unique shape. We perform this for all 3D scenes S, resulting in a set of rasterised scenes from the 3D data, G (see Figure 1). We then select all contiguous polygon clusters from the 2D data P and rasterise them individually, $J_i \subset P$, resulting in a set of high-resolution, binary, rasterised scenes J(see Figure 1). Contiguous regions in the 2D dataset P are defined by polygons p that are connected by shared points or edges. A fuzzy definition could be used if adjacent buildings in a particular dataset do not share points or edges.

Some of the 3D scenes contained arbitrary sections of terrain data. To ensure that G_i is an accurate projection of the buildings, any terrain objects described in a scene S_i should be removed before rasterisation. The area is computed for each set of connected triangles in S_i. Normals are then computed for each triangle face, and the average angle between each pair of adjacent normals, translated to the origin, is computed. Terrain sections tend to have significantly larger areas, and significantly smaller normal variation when compared to man-made structures. A Gaussian is then fitted to this data, and the outlying terrain sections are removed.

[1] `http://sketchup.google.com/3dwarehouse/`

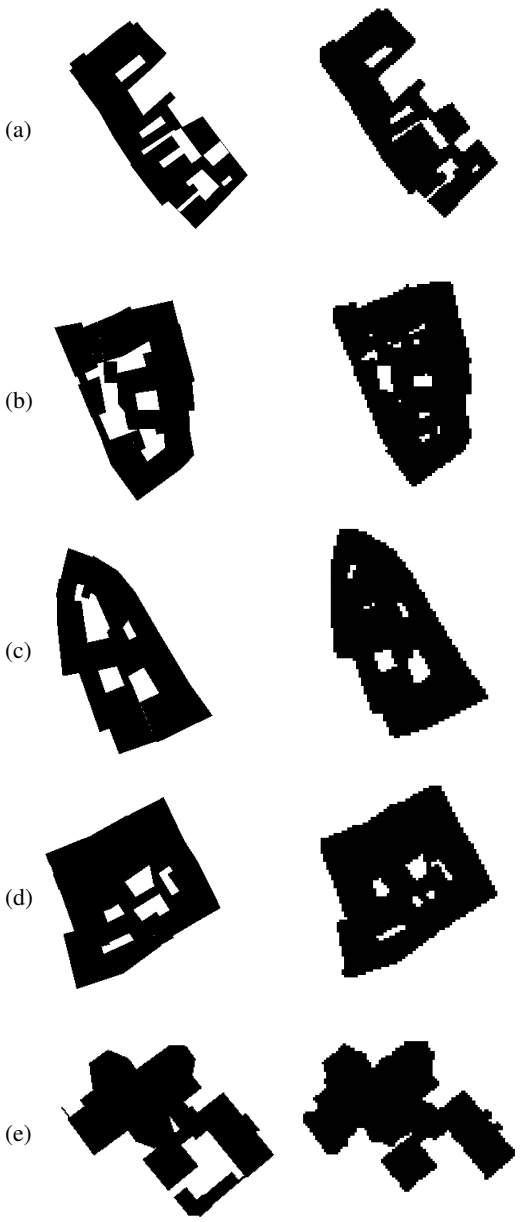

Figure 1: Rasterised building scenes; left images, 3D models (G_i), right images, 2D models (J_i)

2.1.1 Shape Matching The first step in the registration process involves finding the best match for each rasterised 3D scene J_i within the set of all rasterised 2D scenes G. We compute properties that describe the shape of each J_i and G_i well, whilst being invariant to translation, scale, and rotation. We first compute the geometric moments for each shape (Sonka et al., 1993). We also compute the circularity of the shape (Sonka et al., 1993). A description vector $\mathbf{d}$ is computed for each shape, representing translation, rotation and scale invariant moments, and the circularity of the shape. Due to the large discrepancy between the shapes J_i and G_i, we discard all but the first and second moment characteristics, as they reduce the accuracy of the matching process. The shapes are converted to binary images, where a pixel with a value of 1 belongs to the shape, and a pixel of value 0 is the background.

For an binary image of width w and height h, where a pixel co-ordinate is defined as (i, j), we first define the standard shape

moments:

$$\mu_{pq} = \sum_{i=0}^{w}\sum_{j=0}^{h}(i-x_c)^p(j-y_c)^q f(i,j)$$

where x_c and y_c define the centroid of the shape. We then define the normalised un-scaled central moments ϑ_{pq}:

$$\vartheta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^\gamma}$$

where μ_{00} represents the area of the shape, and $\gamma = (p+q)/2+1$. From these moments, we then define the following two moment characteristics:

$$\varphi_1 = \vartheta_{20} + \vartheta_{02}$$

$$\varphi_2 = (\vartheta_{20} - \vartheta_{02})^2 + 4\vartheta_{11}^2$$

We then compute the circularity c of each shape, which can be defined with respect to the shape moments:

$$c = \frac{\sqrt{\mu_{00}}/(\mu_{20}+\mu 02)}{2\pi}$$

We then construct a shape description vector:

$$\mathbf{d} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ c \end{pmatrix}$$

The shape description vector $\mathbf{d}$ is computed for each J_i and G_i, and closest match in G is found for each J_i by simply choosing the smallest Euclidean distance $\sqrt{\sum_{i=1}^{n}(\mathbf{d_i} - \mathbf{d_i'})^2}$, where $\mathbf{d}'$ describes a target shape in G, and n the length of the description vector.

2.1.2 Registration Once the closest match, G_i, for the rasterised image of a 3D scene J_i has been found, the centroids for both shapes are computed, and, J_i, is translated onto G_i. This position is used as the initialisation for the registration process. The feature based registration method introduced by (Borgefors, 1988) is used to register the models. The approach is appropriate for the images used in this work, relying on the matching of line features within the target and query images.

The output of each registration process is a general transformation matrix T, which describes the translation, rotation, and scale. For each mapping $G_i \rightarrow J_i$, the corresponding polygons in P that represent the rasterised set J_i are deleted. The transformation T is then applied to each S_i, which are now aligned to the 2D dataset P. Following this, each scene S_i is segmented into individual buildings according to P. A set of rules is applied to determine basic assignment of polygons s within S_i to a building b, where a b consists of a set of polygons (possibly with textures) within $\mathbb{R}^3$:

1. A disconnected 3D polygon s_i that lies entirely within a polygon p_i is assigned to a building b_i

2. A vertex lying close to the boundary of two polygons p_i and p_j is translated along the internal normal of the boundary edge of the containing polygon, onto the boundary. If the vertex is a member of polygons in both buildings, it is duplicated, with one vertex assigned to bi and one to b_j. We define close by extending the original internal polygon edge normal, and computing the point of intersection with the polygon p_i. Close is then defined as 10% of the distance between the polygon edge and the point of intersection.

3. If a polygon s_i is shared between two buildings, but vertices in s_i are not determined to be close to the boundary, then the polygon is clipped according to a vertical clipping plane defined by the intersecting polygon boundary edge. Vertices are added at the point of intersection for both polygon segments, and assigned to their containing buildings.

The above three rules results in a well-segmented set of buildings that conforms to the building segmentation determined by the accurate 2D ground plan dataset P, which is important for the building annotation discussed in Section **??**. It does not, however, necessarily result in an improvement in the quality of the original 3D geometry. Finally, a set of buildings is defined as $B : \{b_1, \cdots, b_m\} \subset \mathbb{R}^3$, constructed from the application of the above rules to each registered 3D scene S_i. B forms part of the enriched city model C.

2.2 Further Enhancement

The 3D data-set used in this work provides only limited coverage of the city. The remaining buildings that do not have detailed 3D models are therefore extruded from the 2D data-set P. The 2D model P is projected into *DHDN / Gauss Kruger Zone 4*, aligning it with a satellite imagery data-set of the city. For each building, we then approximate a height h using the algorithm described in (Willneff et al., 2005). Each remaining polygon within P is then extruded according to the height h, and a roof mesh is fitted, resulting in a building b, which is added to the set of buildings B belonging to the enriched city model. Rivers and roads removed from the 2D dataset P are then added to the enriched model C (see Figure 2). Finally, a terrain mesh is generated from the digital elevation model, and each building within B is projected onto it.

3 SEMANTIC ENRICHMENT

The process of generating image captions require that thematic attributes, even at a basic level e.g. a name, need to be added to each building. The Keyhole Markup Language (KML) along with most 3D computer graphics formats (VRML, 3D studio Max etc) are only intended to represent 3D objects for visualisation proposes, where detailed semantics of building parts is not catered for. CityGML (Kolbe et al., 2005) however, a relatively new standard, provides an urban landscape ontology suitable to represent the geometry of our city model C alongside thematic attributes about each building and their parts.

Importantly then, a method of adding thematic knowledge about each building and its parts is needed. In (Kumke, 2003) and (Hoegner et al., 2007) 3D models are matched to facts in underlying official municipal and cadastral datasets. However, to the best of our knowledge, no previous work has attempted to automatically enrich a 3D city model (in any format) from freely available Web 2.0 information sources.

3.1 Approach

Our aim in this section is to enhance the city model C with thematic information about each building where available (see Figure 2). The techniques developed are general and applicable to any city model which has been registered to some real world coordinate system.

To find thematic information for buildings b_i, the polygons $p_i \in P$ (in the 2D ground-plan) are matched to point referenced places (or buildings) in Wikipedia, Open Street Maps and the free web

Figure 2: 3D scene registered with 2D ground plan data and annotated using Web 2.0 sources. (a) St. Stephan (church), (b) Obere Pfarre (place of worship), (c) Klosterbrau (pub), (d) Battingerhaus (attraction). 3D model is shown positioned above the 2D ground plan for visualisation purposes.

gazetteer Geonames. Our technique uses each source to determine the name and type of each building e.g. Altes Rathaus which is-a city hall.

Information from georeferenced Wikipedia articles is extracted as RDF[2] triples by the DBPedia project and exposed through a public API. Wikipedia articles are then extracted from DBPedia by querying their SPARQL endpoint. The name of buildings is implied by both the name of the matching entry in DBPedia as well as the title of the corresponding Wikipedia article.

Furthermore, building polygons b_i are matched to building entries in the free online collaborative mapping service Open Street Maps, as well as the free online gazetteer Geonames. These sources do not contain detailed knowledge about building parts, only allowing the acquisition of knowledge about building name and function.

3.2 Matching Issues

Matching building polygons b_i to point referenced locations in ν ranges from simple containment queries (point in polygon checking), to non-trivial cases that involve mapping a single point referenced location to a number of spatially disjoint buildings. These cases, in order of increasing complexity, are now described. From this point onward, the complete set of point referenced building locations ν from Wikipedia / DBpedia (W), Geonames (G) and Open Street Maps (O) is defined as:

$$\nu = \{W, G, O\}$$

Case 1 - Direct Containment In the simplest case, point referenced locations of buildings in ν lie directly inside a polygon b_i. Hence, a simple point in polygon match is applicable, see for example the Cathedral de Bamberg example in Figure 3(a).

Case 2 - One to One Matching Point referenced locations of buildings in ν do not always lie inside a polygon b_i in P as in case 1. Figure 3(b) shows how the Alte Hofhaltung Wikipedia article has been geo-referenced outside of its actual building polygon.

Case 3 - One to Many Matching In a slight alteration to case 2, certain locations in ν can refer to a number of separate buildings. Figure 3(c) shows a Wikipedia reference that refers to a block of 17 separate buildings referred to as part of *Small Venice, a*

[2]The Resource Description Framework - see `http://www.w3.org/TR/REC-rdf-syntax/`

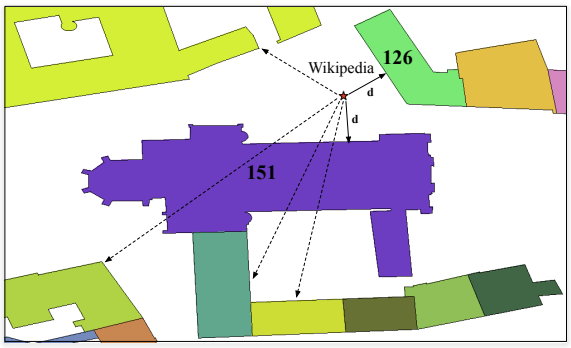

Figure 4: Possible inaccuracies of nearest building match (St. Jakob). The dotting lines represent example distances between the Wikipedia location for St. Jakob and some of the buildings in the ground plan P

former Fischer settlement on the eastern bank of the river Regnitz in Bamberg.

Case 4 - One to One With Many Disjoint Buildings A single building location in ν may map to more than one disjoint building b_i in P that belongs to the same set of buildings. For example, University of Bamberg is comprised of 4 spatially disjoint buildings in this area of the city, see Figure 3(d).

Outcomes: Case 1 is easy to solve with a point in polygon check. Clearly it is possible that, even though the point location is contained inside a single building, it actually maps to more than one building. However, our approach aims to be conservative, where the precision of mapping is more important than recall. In other words, we would rather map to one correct building, than map to many buildings where only a subset are correct. Case 2 could be solved by finding the nearest (in terms of Euclidean distance) building, however in some cases, the nearest building may not always be the correct building - see Figure 4, where building 126 is closer than its correct mapping, building 151. Cases 3 and 4 are, with the information we have, non-trivial. Case 3 would require associating 17 buildings to the same Wikipedia article, while not associating any of the other, often still connected, buildings. An obvious approach here would be to use the registered 3D model to consider occlusion. However, from manual investigation it appears users do not themselves consider occlusion when tagging articles in Wikipedia, or when adding locations in Open Street Maps or Geonames i.e. some references are to occluded buildings, hence this approach would not apply to all cases. Case 4 is a specialization of case 3, which also requires associating a number of nearby buildings to a single wikipedia article, however these spatially disjoint set of buildings are all (semantically) part of the same building.

In this paper we develop two fuzzy mapping function that associates, to a certain degree, each georeferenced information source (the points in ν) with buildings in the 2D ground plan P. By using a fuzzy mapping we hope to overcome some of the issues previously described. The fuzzy mapping functions are described in the sections to follow.

3.3 Fuzzy Mapping Function

In this section we describe the process of linking points in ν to building polygons b_i in P using one of two different fuzzy mapping functions. Once linked to the joined 2D ground plan, this information can be added to the registered 3D city model C.

Here we use the notions of fuzzy relations (Zadeh, 1965) to map locations to buildings. More specifically, we use a fuzzy relation $\mathbf{R} : \nu \times P \mapsto [0,1]$ to map points from ν to buildings in P, where the degree of truth in [0,1] to which the mapping holds is determined using two different fuzzy mapping functions $\mathbf{R}$ as described in sections 3.3.1 and 3.3.2. The fuzzy relation $\mathbf{R}$ forms a new fuzzy set Ω, which is a list containing element and membership degree pairs; $\Omega = \{\{x, y\}, \mathbf{R}_1\}, \{\{x, y\}, \mathbf{R}_2\}, \ldots, \{\{x, y\}, \mathbf{R}_{nm}\}$, where x is a point from the set ν, y is a building from the set P, n is the size of the ν and m the size of P, and $\mathbf{R}$ is their membership degree in [0,1], e.g {{Cathedral de Bamberg, 1191},1}, {{Alte Hofhaltung, 479},0.9}, {{Alte Hofhaltung, 495},0.87}}.

3.3.1 Baseline Fuzzy Mapping The Euclidean distance, d, from point locations in ν to building b_i polygons in P is based on the distance from the point to either, the nearest edge of the building, or to the nearest vertex of the building, depending on which is closer. The baseline fuzzy relationship $\mathcal{R}$ between a point p_i from the set ν, and building b_i from the set P is then computed using a normalised distance measure in [0,1]. That is, by normalising the computed distance d against the maximal distance between the point p and all buildings b_i in P :

$$\mathbf{R}(p, b) = 1 - \left(\frac{d(p, b)}{d_{max}(p)}\right)$$
$$d_{max}(p) = max_{b_i \in P}(d(p, b_i))$$

Relations closer to 1 represent better mappings. All directly contained points (points that lie inside building polygons) have a distance of 0 and hence a degree of membership $\mathbf{R}$ of 1.

Considering multiple evidence across sources: Many points in ν may link to the same building in P. A many-to-one linking can be added as extra evidence for the fuzzy relation $\mathbf{R}$. For example, Figure 5 shows both the Wikipedia point reference and Open Street Maps point reference to the same building (building 151 or St. Jakob's church).

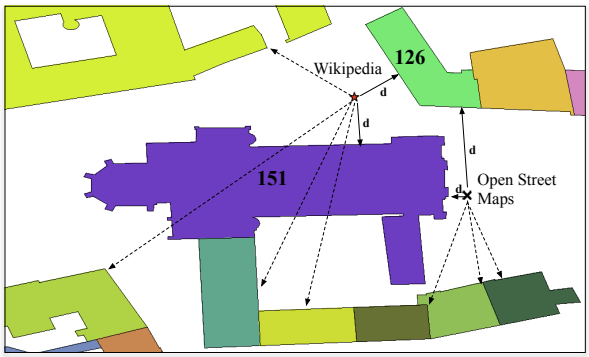

Figure 5: Improving matching by considering multiple evidence (St. Jakob). The dotting lines represent example distances between each location and some of the buildings in the ground plan P

Consequently, the normalised distance fuzzy relationship function is extended to include mappings that consider more than one identical point reference. To identify identical point references in ν, standard and alternative names of each article or POI are matched, using a combined soundex and edit distance fuzzy string similarity measure. Sets of identical references $\nu_1 = \cdots = \nu_n$ are then removed from ν and added to a new set $\nu^{=}$ as tuples $t = \{\nu_1, \cdots, \nu_n\}$ where, for the set $\nu^{=}$, $n \geq 2$. For simplicity,

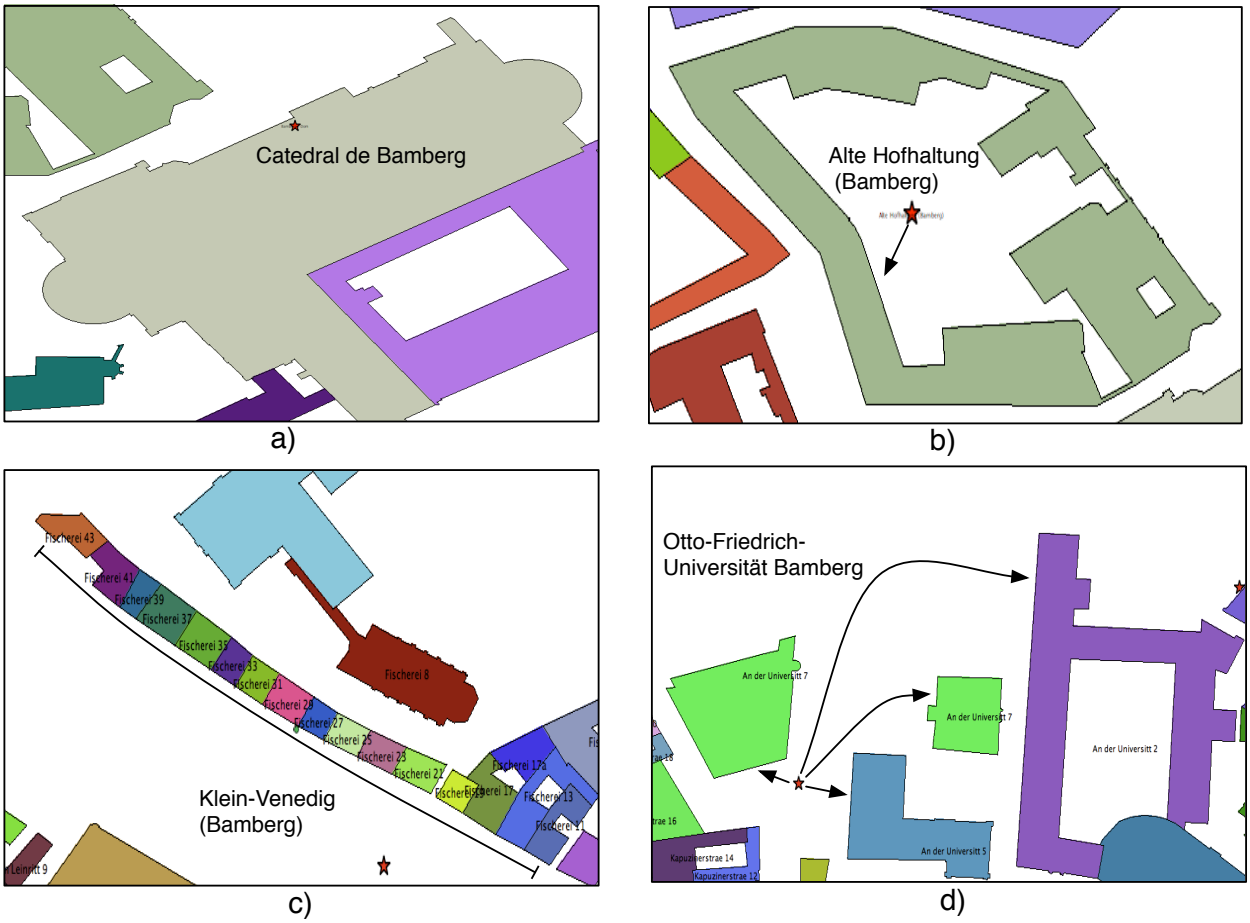

Figure 3: Example mappings between Wikipedia georeferenced articles and buildings in P

we also assume from this point onward that remaining elements of ν are actually tuples t but with only one element i.e. $n = 1$. Hence the total combined set of point references V is then formed from entries in ν and entries in $\nu^{=}$ e.g. $V = \nu^{=} \cup \nu$.

Once V has been established, membership degrees $\mathbf{R}$ in Ω relating point references in V to buildings in P are computed using the normalised mean distance between each evidence point ν_i in a tuple t_i in V (where $0 < i < |t_i|$), and each building. More formally, the relation $\mathbf{R}$ is computed for a tuple t in V and a building b in P as:

$$\mathbf{R}(t,b) = 1 - \left(\frac{d(v_i,b) + \cdots + d(v_n,b)}{d_{max}(t) \times n} \right) \quad \text{, where } n \geq 1$$

Where in this case dmax is used to find the maximum distance between the mean distance of points in a tuple t and all building b_i in P e.g.

$$d_{max}(t) = max_{b_i \in P} \left(\frac{d(v_1,b_i) + \cdots + d(v_n,b_i)}{n} \right)$$

An element membership degree pair $\{\{x,y\}, \mathbf{R}\}$ is then added to the set Ω for each tuple t_i and building b_i by taking the computed degree as the element , the point v_i with minimum distance as element x, and the building b_i as the element y.

Finally, the fuzzy relation function $\mathbf{R}(t,b)$ can be further improved by introducing a common sense heuristic that point references in V will not be over 100 meters from the building(s) they refer to. Hence the function final baseline function $\mathbf{R}_b$ returns a membership degree of 0 for all those buildings outside a 100 meter radius from the minimum distance point v_n in a tuple t_j:

$$d_t(t,b) = \frac{d(v_1,b) + \cdots + d(v_n,b)}{n}$$

$$\mathbf{R}_b(t,b) = \begin{cases} 1 - \left(\frac{d_t(t,b)}{d_{max}(t)} \right) & \text{if } d_t(t,b) < 100 \\ 0 & \text{otherwise} \end{cases}$$

where $n \geq 1$.

3.3.2 Prominent Building Fuzzy Mapping We have also developed an alternative fuzzy membership function which boosts degrees of membership for prominent building shapes. A building classifier is built to detect prominent buildings. Again we build a shape description vector $\mathbf{S}$ for a sample set of buildings from the ground plan P, based on the buildings elongation ϵ (from (Stojmenović and Žunić, 2008)), compactness C (see for example (Lee et al., 2004)) and area A (scale). More formally, a shape description for a building b_i is the vector:

$$\mathbf{S} = \begin{pmatrix} \epsilon \\ C \\ A \end{pmatrix}$$

Shape vectors were learnt for 20 buildings in the ground plan that represented walls, 20 that represented terraced or small buildings and 20 that represented large prominent buildings (0.016% of all buildings in P). The set of learned shapes where added to the training set Lv. The cosine similarity measure is used to match and classify shape vectors for new building shapes against those in Lv.

The new fuzzy membership function denote $\mathbf{R}_{alt}(t,b)$ is then a function of both distance and building shape. More formally, for a tuple t in V and building b in P:

$$\mathbf{R}_{alt}(t,b) = \begin{cases} \frac{W\left(\frac{d_t(t,b)}{d_{max}(t)}\right)^2}{W_{max}(t)} & \text{if } Pb(b) = \text{true} \\ \frac{\left(\frac{d_t(t,b)}{d_{max}(t)}\right)^2}{W_{max}(t)} & \text{if } Pb(b) = \text{false} \end{cases}$$

where $d_t(t,b) < 100$, and $n \geq 1$. Note that if $d_t(t,b) \geq 100$, $\mathbf{R}_{alt}(t,b) = 0$.

Where $Pb(b)$ is a function that takes a building b_i and determines if it is a prominent building by matching its shape description vector against the set of learned shape vectors Lv (as discussed previously). W_{max} is the maximum value (for normalisation) taken from the mean distance of a tuple of points t from V and all buildings b_i in P squared and multiplied with a weighting W if the building b_i is a prominent building.

3.4 Evaluation

For the area of Bamberg, the set V has 53 tuples with only one evidence location v, and 10 tuples with evidence from multiple sources i.e. v_i where $i > 1$. After applying both standard and alternative fuzzy mappings over the grouped set P, the fuzzy set Ω holds mappings between articles in Wikipedia, and entries in Geonames and Open Street Maps and buildings in P. For evaluation, we compare the results of the mapping after applying different thresholds on the fuzzy relationship $\mathbf{R}$, with manual mappings held in a set ψ as defined by a local expert.

For comparison, we first partition the sets Ω and ψ such that each partition ω_i of Ω represents information about a single unique reference v in V, and similarly for each partition ϕ_i of ψ. For each identical partition ω_i and ψ_i (identical in that they are about the same reference v in V) we then compute the following measures for both baseline (standard) and alternative fuzzy relation functions at different threshold levels of $\mathbf{R}$. The first (represented by the columns in Figure 6) is a measure of the number of exact matches between the machine and expert output. That is, how many of the locations have been mapped exactly onto the same number of buildings, without mapping onto other incorrect buildings, as the human expert. The second (represented by the line graph in Figure 6) shows the average (for all v) per threshold level of a combined measure C of mapping accuracy for each unique point location v. More formally, C is defined as:

$$C(v) = \frac{3(1 - NFP(v)) + (1 - NOP(v)) + 2(NA(v))}{6}$$

Agreement A is a count of the number of buildings correctly matched in the machine output with those from the the expert output. Agreement is then normalised (NA) by dividing A by the total number of buildings in the expert output for that partition ϕ. Normalised False positives (NFP) is the count of the number of buildings linked to point references in the machine output that are not contained in the human output, divided by the total number of machine buildings in ω_i. Normalised Omitted Positives (NOP) is the count of the number of buildings linked in the expert output that are not contained in the machine output, divided by the number of buildings in ϕ. Weightings are introduced such that priority is placed on maximising agreement and minimising machine false positives NFP. This is because, it is assumed better to not match all buildings in the machine output to the expert output and have a low number of machine false positives, than match to all buildings in the expert output but also many others not in the expert output - giving erroneous linkage from buildings to a point reference information sources

From the results shown in Figure 6, both the prominent and standard fuzzy mappings over the ground plan P follow a general trend where increasing the threshold increases the combined measure C. The prominent fuzzy mapping has a marginally better maximum combined measure C of 0.704 at a threshold of 0.9, compared to the maximum combined measure C of 0.701 at a threshold of 0.85 for the standard mapping. Furthermore, at this threshold the prominent mapping has a 0.4 (40%) exact match success rate, compared to a maximum of 0.3 (30%) for the standard mapping. Indeed the prominent mapping provides a far better exact match rate than the standard mapping. At the best threshold for the prominent mapping, the average number of NFP is 0.18, which equates to a relatively low 0.29 extra buildings being mapped in the machine output. The average NOP is 0.37 or an average 3.29 buildings in the expert output that were not in the machine output, this is an increase from the best value of 0.14 (1.8 buildings) at a threshold of 0.0. However, as previously stated, we prioritise minimising the average NFP over the average NOP.

Consequently, to achieve the best mappings between location and buildings the alternative mapping $\mathbf{R}_{alt}$ at a threshold of 0.9 should be used. Buildings in P are then linked to information sources at a 0.9 threshold and output as an enriched ground plan P^E . Buildings in the 3D city model C that are registered to the ground plan P^E then inherit the same linkage to information sources.

4 CONCLUSIONS AND FUTURE WORK

In this abstract we have described a method to combine and annotate multiple data sources, in order to produce an enriched 3D city model for use in automatic caption generation for geo-referenced photographs. We describe an approach to robustly register existing 2D and 3D datasets, high-resolution satellite imagery and a digital elevation model, overcoming problems with geometric quality. In addition, web-based methods to enrich the 3D model have been described. Future work will focus on further enrichment methods, and the segmentation of geometric and textural building features.

Current work looks at the extraction of salient building parts, e.g. **St. Peter's Spire**, from the free text content of Wikipedia articles using natural language processing (NLP) and named entity recognition (NER) techniques. We are also focusing on the autonomous correction of inaccurate photograph GPS data using photographic content.

ACKNOWLEDGEMENTS

The research reported in this paper is part of the project TRIPOD supported by the European commission under contract No. 045335.

REFERENCES

Borgefors, G., 1988. Hierarchical chamfer matching: A parametric edge matching algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-10(6), pp. 849–865.

Edwardes, A. J. and Purves, R. S., 2008. Knowledge acquisition for geographic image search. In: GIScience - Fifth International Conference on Geographic Information Science.

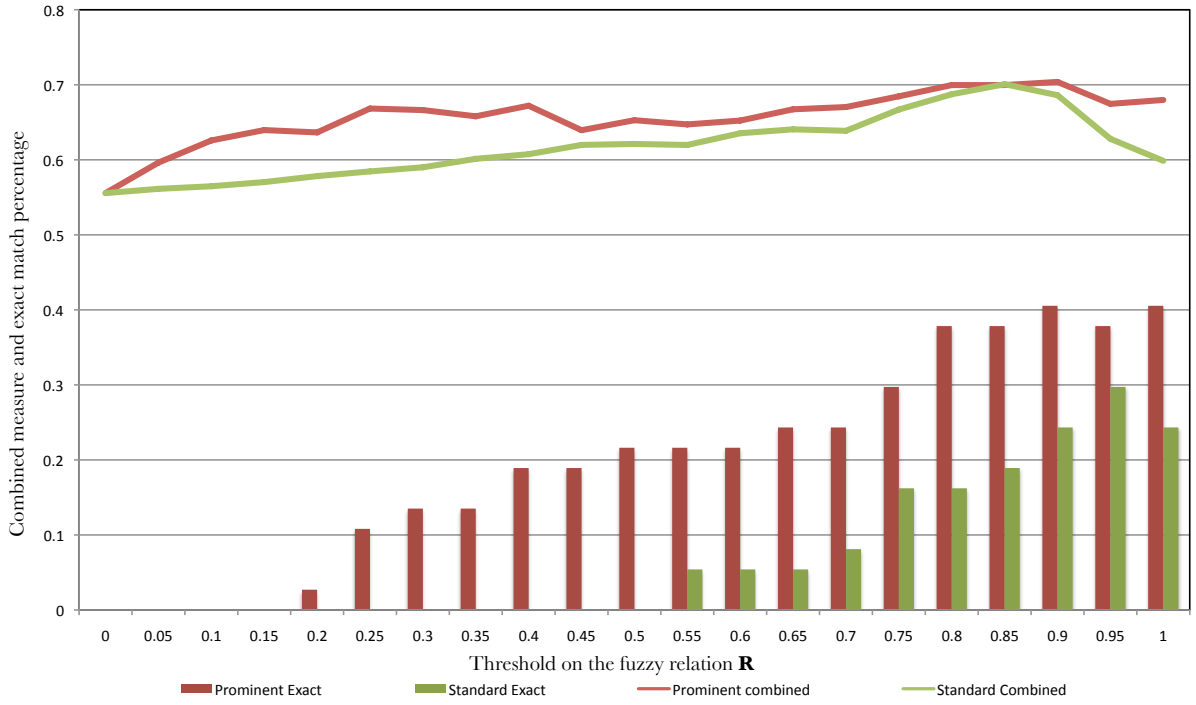

Figure 6: Comparison of the baseline (standard) and prominent fuzzy membership functions

Hoegner, L., Kumke, H., Meng, L. and Stilla, U., 2007. Visualization of building models and factual data integrated by citygml. In: In Proceeding of ICC07.

Kolbe, T., Gröger, G. and Plumer, L., 2005. Citygml – interoperable access to 3d city models. In: In Proceedings of the 1st International Symposium on Geoinformation for Disaster Management, Springer.

Kumke, H., 2003. Salzburg as 3d city model. In: Data Mining and Workflow, AGIT 2003, GISCluster.

Lee, S., Wang, Y. and Lee, E., 2004. Compactness measure of digital shapes. In: Region 5 Conference: Annual Technical and Leadership Workshop, pp. 103–105.

Purves, R., Edwardes, A. and Sanderson, M., 2008. Describing the where—improving image annotation and search through geography. In: In Proceedings of the Workshop on Metadata Mining for Image Understanding.

Sonka, M., Hlavac, V. and Boyle, R., 1993. Image, Processing, Analysis and Machine Vision. Chapman & Hall Computing Series, Chapman & Hall, London, United Kingdom.

Stojmenović, M. and Žunić, J., 2008. Measuring elongation from shape boundary. Journal of Mathematical Imaging and Vision 30, pp. 73–85.

Willneff, J., Poon, J. and Fraser, C., 2005. Single-image high-resolution satellite data for 3d information extraction. International Archives of Photogrammetry, Remote Sensing & Spatial Information Sciences.

Zadeh, L. A., 1965. Fuzzy sets. Information and Control 8, pp. 338–353.