The 13th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2012)
D. Arnold, J. Kaminski, F. Niccolucci, and A. Stork (Editors)

Vectorising Building Footprints From Historic Maps

P.G. Brown[1], S.D. Laycock[1] and A.M. Day[1]

[1]School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK
{philip.brown, s.laycock, a.day}@uea.ac.uk

Abstract

Historic maps provide fascinating insights into the growth and development of cities over large time periods, and often the places they document have changed dramatically since their creation. However, rasterised image data is limited in that it is not easily manipulatable, for either analysis or presentation of data, and it is not easy to use the data as a basis for reconstructions. If it were possible to extract vector outlines for building footprints from the map then these limitations would be removed, and the vector outlines could easily be used as a basis for three dimensional models representing the buildings. Manually defining building outlines from maps is a laborious process and so this paper investigates the use of image processing techniques to enable a semi-automatic process to detect vector outlines of buildings from a variety of maps and to optimise the output for use as a basis for creating large scale three dimensional reconstructions.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Historic maps and sketches of towns and cities offer invaluable documentation of the layout and dynamic of settlements as they have changed through the years. In the UK, Ordnance Survey maps detailing building locations have been produced since the 18th Century. In the USA Sanborn maps have been produced for American cities since the mid-19th Century, recording building footprints and other details for the purposes of assessing fire insurance liability. Historic maps also exist for a wide range of purposes and from a wide range of disparate sources e.g. maps constructed by local governments for the purposes of assessing land ownership and usage, or maps commissioned for military use.

Some websites [MAP12a] [MAP12b] allow real-time comparison between historic raster images and modern data by use of an overlay or side-by-side comparison. However this type of comparison is limited, and vectorising the data would provide multiple benefits such as fully customisable visualisation (allowing use of colours and shading styles), providing scale-independent quality, easy editing of data, and an ability to attribute properties to individual buildings. Producing vector data for buildings from a variety of maps with different styles would also allow those separate data sets to be displayed in an homogenous fashion. Additionally once vector footprints have been obtained they can easily be used as the basis for further reconstructions using software products such as CityEngine [MWH*06]. Figure 1 shows an example of some vectorised building footprints that have been extruded into simple three dimensional block shapes to create a visualisation of selected buildings from an historic map. Alternatively, a collection of three-dimensional models may be created by digital artists in order to populate the scene [LLD09].

In order to obtain vector footprints to represent buildings from raster images, labour intensive manual digitisation is often required using software tools such as ArcGIS [ARC10]. In this paper, techniques are presented which use image processing techniques to semi-automatically extract building footprints from digital images of archive maps in order to produce vector outlines of the buildings. The next section covers relevant previous work.

2. Previous Work

There are many methods for creating three dimensional models of cultural heritage sites where buildings and objects still exist, either partially or fully. Two prominent technologies for creating three dimensional models of existing his-

DOI: 10.2312/VAST/VAST12/017-024

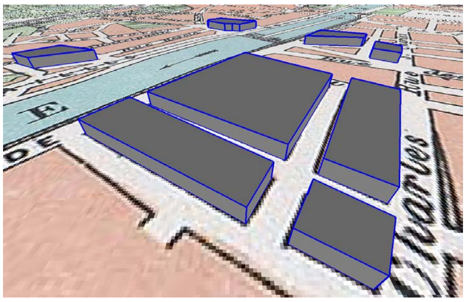

Figure 1: *This image shows a map with vectorised building footprints that have been extruded into simple three dimensional block shapes in order to create a visualisation of selcted buildings of a city from an historic map.*

toric buildings and artefacts are photogrammetry [KHN04], and laser scanning [MT00] [STY*03] [ZYC06], both of which have been used to record and document cultural heritage sites. However, in circumstances where digitising a physical object is not possible because it no longer exists, alternative methods are required. This paper considers town and city areas that have undergone significant changes and alterations, and so data for their recreation must be acquired from historic sources.

To tackle the problem of recreating large city environments, Muller et. al, [MWH*06] used procedural modelling techniques to reconstruct the city of Pompeii using a structured grammar. However, only a limited portion of the buildings were aligned to known building footprints, whilst the majority were procedurally generated based on the street network using techniques presented in [PM01]. Guidi et. al, [GFDS*05] used a special metrology Laser Radar to scan a large plaster-of-paris model of Imperial Rome that was created in the 1970s in order to generate a huge 3D model of the city.

Building footprints have been extracted from high-resolution aerial images using a variety of methods [FKL*98] [PZL05] [DH09]. However these techniques are not directly applicable to historic maps, which have a number of different issues e.g. lower quality, notations on the map (road names etc). High resolution aerial images are also obviously only available for recent time periods, whereas historic maps cover a significantly larger span of history.

The automatic extraction of building footprints from archive maps of Kawagoe, Japan during the Edo period was investigated by Suzuki and Chikatsu, [SC03]. The method presented recovered rectangular footprints only, by generating templates and aligning these to corners detected in the image. Laycock et. al, [LDD08], overcame this limitation by considering recovering building footprints represented by non-intersecting closed polygons. However, this introduces further complications, since a polygon extracted from an image may have unit length edges. These require refinement such as the removal of near collinear points or the complete replacement with the largest oriented empty rectangle that can fit within the polygon.

Shimizu and Fuse, [SF03] used a piecewise linear affine transformation strategy to align multiple maps from different periods in order to analyse changes in land-use over time. Building on this, Laycock et. al, [LBLD11], used multiple maps of the same area over different time periods to aid in the extraction of building footprints to allow analysis of changes to an urban environment.

Baily, [Bai07] investigated the use of image processing techniques to vectorise information from land usage maps, relying primarily on colour differences to identify areas and to eliminate clutter from maps such as contour lines and text. However, this work was primarily concerned with the area covered by different colour values, rather than accurately converting the shape of the raster image to vectors.

Willis et al. [WSG09] vectorised architectural plans of castles and fortifications in order to generate 3D recreations of the buildings. The approach vectorised the plans by skeletonization and then fitted curve fragment shape models for each graph edge. The authors acknowledge that the approach is not suited for cases when the building is low-resolution or when the building boundary is not thick and distinguishable from other annotations in the image, or cases where buildings neighbour one another, all of which are often problems with historic map sources.

Work has been done to investigate the vectorisation of mechanical drawings [CLD96] [DHR91]. Hilaire and Tombre, [HT06] vectorised mechanical drawings by skeletonizing the image and then segmenting the result into primitives (lines and arcs). Text removal techinques by Tombre et al. [TTP*02] were used to remove text that was not to be vectorised. Techniques for accurate vectorisation of mechanical drawings such as those discussed have produced good results, however the requirements for extracting building footprints differ. Fully accurate vectorisation of all lines is not what is desired, and attempts should be made to vectorise only the relevant details. For example, vectorising every line present in Figure 2 would produce a lot of extra information beyond building outlines. In cases such as buildings shaded by cross-hatching, accurately vectorising all the lines from an unprocessed image would in some cases vectorise the cross-hatching and as such produce unusable results.

Commercial tools offer raster-to-vector methods, such as the ArcScan extension to ArcGIS which allows automatic vectorisation of simple block shapes from greyscale images, [ARC12] , AutoCAD Raster Design which offers vectorisation via line tracing [AUT12] and Vextractor which uses a semi-automatic approach to vectorise various types of raster images [VEX12]. These tools can produce good results when vectorising lines. However, as mentioned above,

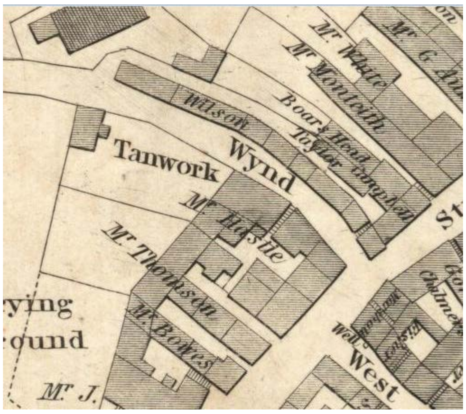

Figure 2: *An example map with many distractions such as lines and markings that are not relevant to the building outlines. Accurately vectorising all lines in the image would be undesirable for the stated goal of extracting building footprints. Note also that text intersects the edges of buildings and lines frequently.*

the focus of this paper is not to convert all the lines in an image to a vector, but to extract only building outlines as vectors while automatically ignoring other details on the map such as boundary lines and other markings wherever possible. Additionally, general purpose tools such as these often do not generate vectors with the desired topology i.e. one closed loop per building, see Figure 3. They are also not optimised for historic sources such as maps which can be hard to obtain in high resolution, tend to have many imperfections and distractions, and use a variety of different styles to represent buildings.

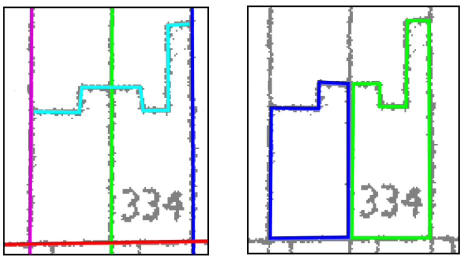

Figure 3: *A pair of example building footprints shown with arbitrary connectivity with each individual vector shown in a different colour as may result from some vectorisation routines (left) and in the desired configuration with only building footprints presented as connected loops (right).*

The techniques employed in this paper are designed to work around the distractions and imperfections found in historic maps in order to vectorise building footprints from a variety of map styles in a semi-automatic process where user interaction is minimised.

3. Footprint Extraction

When attempting to extract vector outlines of multiple buildings from raster images, an automatic or semi-automatic process is needed to avoid the time consuming manual alternative of individually sketching each building. By combining multiple image processing techniques in a specific order we can decompose the map to remove or minimise distractions such as labels, markings, and non-building entities, allowing the extraction of the basic shapes of buildings from the map. This section describes the process used to allow a semi-automated identification of the footprints of buildings from a sample of archive maps with varying styles.

3.1. Image processing

The aim of the image processing is to transform the image to show only the edges of the buildings. Due to the nature of maps, a simple routine such as a Canny edge detect [Can86] would produce unusable results (see Figure 4). For images where shading is used on buildings, the intention is to create a binary mask with the areas representing buildings shown in black and the rest of the image reduced to white, such that each building footprint that is to be identified is represented as a solid black block with as few other markings as possible (Figure 5). A Canny edge detect can then be run on this image, and the contours that result are then vectorised. For buildings with no shading, some processing is done to optimise the outlines for footprint extraction and a Canny edge detect is also used.

Figure 4: *A map with cross-hatching shading style shown in its original form (left). Running a standard Canny edge detect on this image highlights far more than simply the building edges (centre). However if the map is processed to remove distractions before edge detection, the image can be reduced to the relevant edges (right).*

Many historic maps have different styles which will affect the order and combination of the image processing steps. Different resolutions and building sizes will affect the values required as inputs to the processing steps.

The order in which to combine specific image processing techniques depends on the type of image that is to be processed. Historic maps tend to use one of three main styles of shading on the buildings: no shading, cross-hatched shading, and solid shading (typically grey). The order and combination of processing varies for each of the three styles. The user

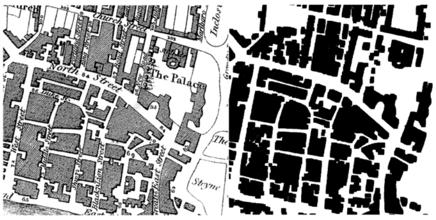

Figure 5: *An historic map shown before (left) and after (right) processing to reduce the image to simple black blocks representing buildings.*

is presented with a preset for each style which automatically applies all the steps needed. The user may adjust parameters to the preset using sliders. Firstly, to accommodate for the fact that buildings may be represented by vastly differing numbers of pixels depending on the map scale and the image resolution, the size of the image kernel used for steps that involve actions such as erode and dilate may be changed. A second slider controls the value used for thresholding an image, as this will vary according to each map's brightness and the colour/shading used to represent the buildings. As the user moves the sliders, a preview is updated in realtime, showing the user only the cumulative result of the final step. This allows non-expert users to apply and tweak complex image processing routines. The following paragraphs describe the processes that constitute each preset.

For all styles of maps, the first step of the image processing is to resample the image to twice its original size. Obviously this does not increase the quality of the image, but several of the methods used in subsequent steps involve passing a kernel over the image and the extra spacing between pixels allows greater control and creates better results.

The most common issues facing building extraction are that there are many features on the map which are not buildings, and sometimes they intersect building outlines or join together two otherwise separated buildings, as seen in Figure 2. Examples of these features include writing and markings on the map such as road lines, trees, street names and other labels. Figure 6 shows an example. We can take advantage of the fact that these features are thin, generally a few pixels wide, in comparison to buildings which are large blocks. Using a combination of erode and dilate operators, it is possible to remove the small unwanted features such as text and markings on the map without significantly disrupting large features, such as buildings. This method is applicable for all map styles.

3.1.1. Solid shading

Maps with solid shading of the buildings (in older maps this is typically grey, although many colour maps also exist) are advantageous when compared to maps with no shading in that buildings are of a different colour/intensity value to the background. This simplifies the thresholding step by allowing us to simply transform buildings into binary blocks via thresholding.

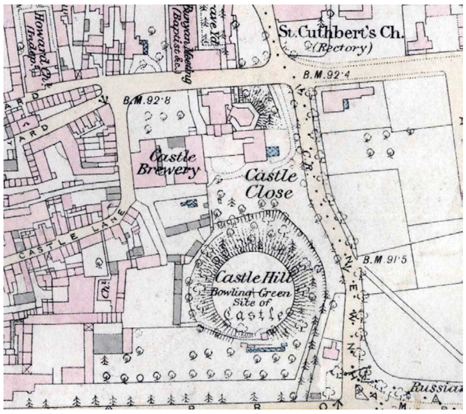

Figure 6: *A portion of a map showing a number of markings that are not relevant to building outlines such as trees, field boundaries, and marking of geological features.*

3.1.2. No shading

For maps where the buildings have no shading (typically white or pale background, with only black lines representing building outlines and other features), thresholding in order to remove everything other than buildings will not work as the majority of the map will be the background colour regardless of whether it is a building or not. Dilating the image slightly can be used to close small gaps in lines that may affect the edge detection results, and an erosion can then return lines to roughly their original size. One notable problem with images with no shading is that all "background" areas which have edges bordering them will appear to be buildings. To mitigate this, a post-processing step can be run, after vector outlines have been obtained, to remove extremely large 'buildings' which have resulted. However, there may still be some manual clean-up required to remove non-building shapes such as gardens (see Figure 7).

3.1.3. Cross-hatching

In images with buildings shaded with cross-hatching, the lines that create the cross-hatching present significant problems to building outline detection. In historic maps, the cross-hatching is usually not of uniform direction or spacing. This therefore makes it impractical to detect the cross-hatching pattern, as it is indistinguishable from other lines on the map e.g. the building outlines themselves. It was determined that the best way to process these images involves eroding the image so that the cross-hatch lines join up, forming the solid block of the building. Processing can then carry on as for solid shaded buildings. One notable drawback is

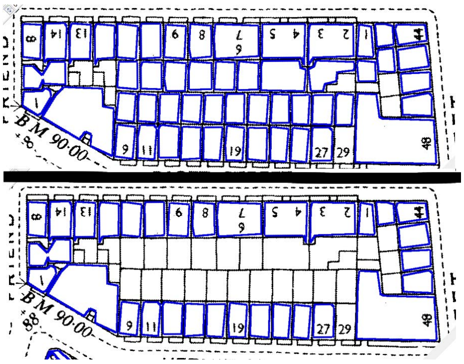

Figure 7: *Footprints (shown in blue) extracted from a map with no shading. This figure demonstrates that in maps with no shading, shapes that are detected are not necessarily desirable if building footprints are the focus of interest. The map is shown with all of the originally detected footprints (top), and with the shapes representing gardens manually removed (bottom)*

that where buildings are subdivided (see Figure 8), the subdivisions will be lost and only the outer building shape will be detected. This could be later corrected manually if required. Maps with incredibly sparse cross-hatching may still prove problematic, or may benefit more from the approach for solid shaded buildings.

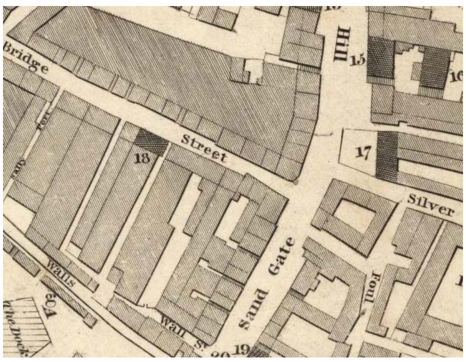

Figure 8: *An example of buildings shaded with cross-hatching and featuring subdivisions of buildings. With the processing method presented in this paper, these may not be represented in the final extraction.*

3.2. Contour detection

The contours from the final generated binary image are then detected using OpenCV's contour detection method [Bra00]. The outlines returned have one point per pixel, so for a building consisting of 50 pixels in the image, a 50-point vector will be returned. For the majority of use cases it can be expected that simplified shapes are necessary or desirable. Therefore a number of simplification routines are run on the initial results in order to reduce the number of points in a single footprint whilst keeping the results as accurate to the original as possible.

3.3. Contour post-processing

Initial simplification is carried out by means of the Douglas-Peucker algorithm [DP73]. Two points at opposite sides of the footprint are selected to define the start point for this algorithm, and then the furthest point from the original shape is added to the line. The process repeats recursively until the distance between the line being processed and the new point to be added is below a certain threshold. Once this simplified shape has been obtained, some domain knowledge can be applied to optimise the process for buildings. Points that are extremely close together can be merged into a single point based on the expectation that buildings will not have implausibly small walls and two such points close together have likely arisen from aliasing from the original binary image. Tests are also performed to test for shapes with extremely small areas which may have arisen due to artefacts of the processing or unusual features on the map, and also there are tests to ensure no self-intersecting shapes have been created.

4. Results

The techniques presented here can be considered to be a success if they reduce the time needed to manually obtain vectorised footprints from an image whilst still providing a reasonable level of accuracy. In order to establish if this is the case, the footprints obtained from the semi-automatic processes presented here were compared to a fully manual set of results obtained by a user tracing each building footprint. A varied set of images from different sources and of differing resolutions representing each of the three main styles of map were tested, and the details of these are provided in Table 1.

Figure 9 shows an example of a map at several stages throughout the image processing step, showing how the processing affects the image and reduces it to the desired components. Figure 10 shows the result of footprint extraction on an image of Paris from 1879.

For both the manual and the semi-automatic approaches, the time taken to obtain a set of results was recorded, as well as the final building count. For the semi-automatic approach, the timing is broken down into two sections: the time taken to remove any non-building shapes that have been detected and manually trace buildings that were missed, and the time for manual editing of results in order to correct any large mistakes found in this base set. (The time required to execute the image processing routines and contour detection method is negligible in this context, at around 3 seconds for a 10 megapixel image, and so it is not detailed here).

Table 2 shows that the semi-automatic process provides a

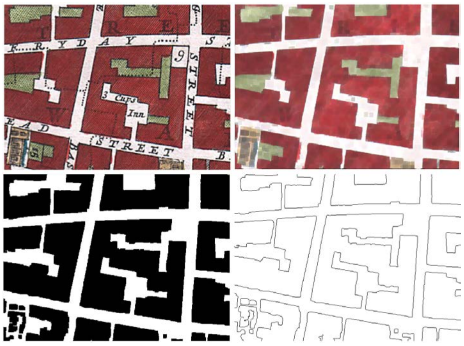

Figure 9: *A map at various stages of image processing. The original map (top left) shows many distractions such as text that we wish to be removed. After using OpenCV's close gaps operator (top right), small elements have been removed whilst buildings remain largely unaffected. The image is then thresholded (bottom left) and edge detected (bottom right).*

Figure 10: *A set of building footprints (shown in blue) extracted from a map of Paris from 1879. The automatic process returned 1123 valid building shapes for the whole map. The results shown here are before any manual editing has taken place.*

significant time saving over a fully manual extraction. After the initial extraction of footprints, time is required to manually edit the results. Time is taken to remove extraneous shapes which are commonly caused by map markings being mistaken for buildings (e.g. thick lines of text), or as seen in maps with no shading, shapes are correctly detected but we are not interested in them (e.g. London 1955 map where gardens are drawn in the same style as the houses they belong to). Buildings which may have been missed during the extraction are added (these are generally small buildings, or buildings with large text covering a significant portion of their surface area). Extracted buildings are edited when there are obvious errors, commonly caused either by two buildings being joined together due to thick areas of text bridging the gap between them, or by issues with the map such as inconsistent shading. The London 1720 map exhibited a particularly high manual extraction time given the number of buildings present, this was due to the intricate nature of the building shapes requiring extra time to accurately sketch them manually (e.g. one building featured 98 vertices).

Map Title	Shading Type	Image Resolution	Num. of Buildings
Par. 1879	Solid	3888x2608	1212
Lon. 1720	Solid	850x1434	90
Lon. 1955	None	2531x2603	333
Bri. 1822	Cross-hatched	2648x1800	235
Dag. 1822	Cross-hatched	1896x876	38

Table 1: *A summary of the maps used for testing in this paper, giving details such as the style of shading and number of buildings represented.*

Map	Semi-auto. Time (m)		Manual
Title	Rem./ Add.	Edit	Time (m)
Par. 1879	23:45	12:00	180:00
Lon. 1955	8:30	7:00	47:00
Bri. 1822	8:15	6:30	40:00
Dag. 1822	1:00	2:00	10:00
Lon. 1720	1:40	3:00	28:00

Table 2: *A table showing timings for the extraction comparing the time required to manually sketch all of the buildings on a map compared to the semi-automatic process. The semi-automatic times detail time spent removing erroneous shapes and adding shapes that were missed (Rem. / Add.) and editing automatic shapes to correct obvious errors (Edit).*

In order to evaluate the accuracy of the results of the semi-automatic extraction, we compare them to a set of results known to be correct. The correct set of results is generated by a user drawing around each building on a map. By comparing corresponding buildings from this set and the extracted results, we can determine error values that show the level of accuracy of the automatic process.

Each pair of corresponding footprints is rasterised to a pixel grid so that an image-based metric can be used to determine the distance between two feature vectors, each comprised of a set of n pixels representing the building footprints, in the form $[(x_1,y_1)\dots(x_n,y_n)]^T$. The distance metric used is based upon central moments. Hu [Hu62] proposed a set of seven moment invariants, which are invariant to translation, scale and rotation. The moment invariants are based upon scale-normalised centralised moments of up to order 3. The invariants are calculated and the difference between two footprints is obtained by calculating the Euclidean distance between two sets of moment invariants.

Table 3 displays the distance values that were calculated when comparing the two building sets for each map. The average distance is shown as well as the maximum and minimum in order to give a complete picture of the accuracy.

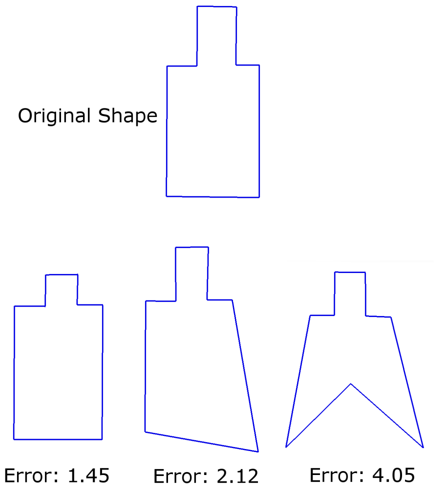

Figure 11: *An example building compared with three variants. The distance between the shapes has been calculated and shown. Higher error values are a result of large differences between the two shapes, whereas comparing a shape to itself would result in an error value of zero because it is a perfect match.*

Map Title	Manual vs. Semi-auto. Avg.	Min.	Max.
Paris 1879	0.80	0.01	2.04
London 1955	0.61	0.02	1.92
Brighton 1822	1.83	0.07	3.43
Dag. 1822	0.32	0.01	1.24
London 1720	0.22	0.01	3.77

Table 3: *A table showing the accuracy of buildings extracted by the automatic process. The results are compared with manual tracings of the buildings, which is used as an accurate result to test against. The distance between the two shapes is then calculated. Average error values are reported as well as the minimum and maximum values.*

Figure 11 shows some example shapes and three altered versions of the same shape underneath, alongside the distance values calculated, in order to give some context to the results presented. Higher error values are a result of large differences between the two shapes, whereas comparing a shape to itself would result in an error value of zero because it is a perfect match. Only unedited shapes from the original extraction were used for the results, as checking accuracy of manually edited or created shapes would be redundant and misleading.

The results displayed in Table 3 show that the results of the extractions generally have a good level of accuracy. The London 1955 map has a high level of accuracy, although as seen in Table 2 some time was needed to remove extraneous shapes detected as a result of the map not having any shading to help identify building shapes. The Brighton 1822 map has a higher level of average error and this can be attributed to the fact that dealing with cross-hatched shading requires more aggressive image editing than other types and the map contains buildings of fairly intricate shape which suffer as a result.

The Dagleith 1822 map has low error values, however it is important to note that building subdivisions were lost during this extraction meaning neighbouring buildings were extracted as a single conjoined footprint. The manually sketched shapes that it was compared to generate the error values were drawn in the same way. Therefore although the error values are low, information has clearly been lost.

5. Conclusions

The results presented in this paper show that semi-automatic extractions of building footprints from various styles of historic maps can save significant time compared to a fully manual vectorisation process. For use on large-scale observations of a large area containing hundreds of buildings the errors are minor and the results are suitable for use as a basis for a 3D visualisation of the scene. However factors such as the style of building representation and the resolution of the map can affect the accuracy of the results. Images where buildings are represented by only a few pixels will not generate satisfactory results.

Manual vectorisation will provide the greatest accuracy, however in most cases it will be more time consuming and in some cases significantly so. For example in the London 1720 map, the semi-automatic processing time was only 17 percent of that of a fully manual extraction due to the complex nature of building shapes which increased the manual sketching time. It must also be considered that some historic maps may not be perfectly accurate in themselves due to the limited technologies available at the time.

Further work to improve upon the methods here could include methods for making the extraction more robust. For instance, a method could be devised to combine results from two different procedures of the same map. This might be of use in situations where one pass does not provide adequate results e.g. an area of the map has sub-standard image quality, the style of building representation changes at different points, or buildings of vastly different sizes may benefit from different parameters being used. In order to decrease the amount of manual intervention required, a user could simply highlight one building from the map and an algorithm could attempt to determine the style of the building representation on the map, and as such the optimal settings to use for the image processing steps.

5.1. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231809.

References

[ARC10] ArcGIS, accessed 2010. URL: http://www.esri.com/software/arcgis/index.html. 1

[ARC12] Arcscan tutorial, accessed 2012. URL: http://help.arcgis.com/en/arcgisdesktop/10.0/pdf/arcscan-tutorial.pdf. 2

[AUT12] Autocad raster design, accessed 2012. URL: http://usa.autodesk.com/autocad-raster-design/. 2

[Bai07] BAILY B.:. 209–223. URL: http://www.e-perimetron.org/Vol_2_4/Vol2_4.htm. 2

[Bra00] BRADSKI G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000). 5

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, 6 (June 1986), 679–698. URL: http://dx.doi.org/10.1109/TPAMI.1986.4767851, doi:10.1109/TPAMI.1986.4767851. 3

[CLD96] CHEN Y., LANGRANA N. A., DAS A. K.: Perfecting vectorized mechanical drawings. *Computer Vision and Image Understanding 63*, 2 (1996), 273 – 286. URL: http://www.sciencedirect.com/science/article/pii/S1077314296900193, doi:10.1006/cviu.1996.0019. 2

[DH09] DORNAIKA F., HAMMOUDI K.: Extracting 3d polyhedral building models from aerial images using a featureless and direct approach. In *Machine Vision Applications* (Yokohama, JAPAN, May 2009). 2

[DHR91] D. P., H.F. L., R. J.: Graphic features extraction for automatic conversion of engineering line drawings. *ICDAR91* (1991), 533–541. URL: http://www.primaresearch.org/ICDAR1991/. 2

[DP73] DOUGLAS D. H., PEUCKER T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10* (October 1973), 112–122. doi:10.3138/FM57-6770-U75U-7727. 5

[FKL*98] FISCHER A., KOLBE T., LANG F., CREMERS A., FÖRSTNER W., PLÜMER L., STEINHAGE V.: Extracting buildings from aerial images using hierachical aggregation in 2d and 3d. *Comput. Vis. Image Underst. 72*, 2 (1998), 185–203. 2

[GFDS*05] GUIDI G., FRISCHER B., DE SIMONE M., CIOCI A., SPINETTI A., CAROSSO L., MICOLI L., RUSSO M., GRASSO T.: Virtualizing ancient rome: 3d acquisition and modeling of a large plaster-of-paris model of imperial rome. *Videometrics VIII 5665* (2005), 119–133. 2

[HT06] HILAIRE X., TOMBRE K.: Robust and accurate vectorization of line drawings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 6 (june 2006), 890 –904. doi:10.1109/TPAMI.2006.127. 2

[Hu62] HU M.: Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory 8* (1962), 179–187. 6

[KHN04] KIM Z., HUERTAS A., NEVATIA R.: Automatic description of complex buildings with multiple images. In *Computer Vision and Image Understanding* (2004), vol. 96, pp. 60–95. 2

[LBLD11] LAYCOCK S., BROWN P., LAYCOCK R., DAY A.: Aligning archive maps and extracting footprints for analysis of historic urban environments. *Computers & Graphics 35*, 2 (April 2011), 242–249. 2

[LDD08] LAYCOCK R., DRINKWATER D., DAY A.: Exploring cultural heritage sites through space and time. *ACM Journal of Computing and Cultural Heritage 1*, 2 (2008), 1–15. 2

[LLD09] LAYCOCK R., LAYCOCK S., DAY A.: Reconstruction of large cultural heritage sites from archived maps. In *VAST* (Malta, 2009), pp. 41–48. 1

[MAP12a] Historic map explorer, accessed 2012. URL: http://www.historic-maps.norfolk.gov.uk/mapexplorer/. 1

[MAP12b] Manchester historical maps, accessed 2012. URL: http://manchester.publicprofiler.org/beta/index.php. 1

[MT00] MORGAN M., TEMPFLI K.: Automatic building extraction from airborne laser scanning data. In *Proc. 19th ISPRS Congr.* (2000), pp. 616–623. 2

[MWH*06] MUELLER P., WONKA P., HAEGLER S., ULMER A., GOOL L. V.: Procedural modeling of buildings. *In Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics (TOG) 25*, 3 (2006), 614–623. 1, 2

[PM01] PARISH Y., MUELLER P.: Procedural modeling of cities. In *In Proceedings of ACM SIGGRAPH 2001, ACM Press / ACM SIGGRAPH* (2001), ACM Press, pp. 301–308. 2

[PZL05] PENG J., ZHANG D., LIU Y.: An improved snake model for building detection from urban aerial images. *Pattern Recognition Letters 26(5)* (2005), 587–595. 2

[SC03] SUZUKI S., CHIKATSU H.: Recreating the past city model of historical town kawagoe from antique map. *Archives of Photogrammetry and Remote Sensing XXXIV-5/W10* (2003), 1–6. 2

[SF03] SHIMUZU E., FUSE T.: Rubber sheeting of historical maps in gis and its application to landscape visualization of old-time cities: focussing on tokyo of the past. In *8th International Conference on Computers in Urban Planning and Urban Management* (2003). 2

[STY*03] STUMPFEL J., TCHOU C., YUN N., MARTINEZ P., HAWKINS T., JONES A., EMERSON B., DEBEVEC P.: Digital reunification of the parthenon and its sculptures. In *VAST 2003* (Brighton, UK, 2003), pp. 41–50. 2

[TTP*02] TOMBRE K., TABBONE S., PÃĽLISSIER L., LAMIROY B., DOSCH P.: Text/graphics separation revisited. In *in: Workshop on Document Analysis Systems (DAS* (2002), Springer-Verlag, pp. 200–211. 2

[VEX12] Vextractor, accessed 2012. URL: http://www.vextrasoft.com/vextractor.htm. 2

[WSG09] WILLIS A., SUI Y., GALOR K.: Parsing architecture within plan drawings with application to medieval castles and fortresses. In *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage* (2009), The Eurographics Association, pp. 17–24. 2

[ZYC06] ZHANG K., YAN J., CHEN S.: Automatic construction of building footprints from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing 44(9)* (2006), 2523–2533. 2