

Metadata of the chapter that will be visualized in SpringerLink

Book Title	Augmented Reality and Virtual Reality	
Series Title		
Chapter Title	Augmented Reality for Mobile Devices: Textual Annotation of Outdoor Locations	
Copyright Year	2018	
Copyright HolderName	Springer International Publishing AG	
Corresponding Author	Family Name	Larabi
	Particle	
	Given Name	Slimane
	Prefix	
	Suffix	
	Division	Faculty of Electronics and Informatics
	Organization	USTHB University
	Address	Bab Ezzouar, Algeria
	Email	slarabi@usthb.dz
Abstract	For textual annotation of outdoor locations we propose in this paper an augmented reality method for mobile devices based on orientation and GPS measurements and uses a circular string of identifiers as data structure in order to determine what places are seen by the mobile camera. The proposed solution is generic and can be applied in all areas of the world if the required data are available (computed in off-line). The proposed solution has been implemented and served in our university campus for new students to discover all places.	
Keywords (separated by '-')	Augmented reality - Mobile device - Textual annotation - Outdoor location - Circular string of identifiers - Convex hull	



Augmented Reality for Mobile Devices: Textual Annotation of Outdoor Locations

Slimane Larabi

Abstract For textual annotation of outdoor locations we propose in this paper an augmented reality method for mobile devices based on orientation and GPS measurements and uses a circular string of identifiers as data structure in order to determine what places are seen by the mobile camera. The proposed solution is generic and can be applied in all areas of the world if the required data are available (computed in off-line). The proposed solution has been implemented and served in our university campus for new students to discover all places.

Keywords Augmented reality · Mobile device · Textual annotation · Outdoor location · Circular string of identifiers · Convex hull

1 Introduction

The aim of AR is the intertwinement of digital objects, texts and information with the “original” environment (Liberati 2016). Many AR applications have been developed for mobile devices and devoted to many subjects such as text document, adoption behavior, tourism, events, gamification, smart city, retail and cultural heritage.

Augmenting with text has attracted the interest of many researchers and many systems have been proposed such as the system of Ryu and Park (2016) able to detect text documents in real scenes, to estimate their relative 3D poses to the camera, and to augment them with virtual contents.

Annotation in AR systems has been studied in Wither et al. (2009) showing its usefulness either for adding information in a direct or indirect manner. Many systems of augmented reality (AR) devoted for annotation of building, subway maps, and a museum have been proposed (e.g. Eaddy et al. 2004; Schmalstieg and Wagner 2007; Wither et al. 2009). For the proposed systems, many authors asserted

S. Larabi (✉)

Faculty of Electronics and Informatics, USTHB University, Bab Ezzouar, Algeria
e-mail: slarabi@usthb.dz

28 than this orientation measurement is often deviated around tens degrees because of
29 noise, jitter and temporal magnetic influences (Langlotz 2011) and the presence in
30 densely occluded urban environments which decreases the GPS accuracy, this
31 imply that annotations will simply appear on the wrong location.

32 In other side, using computer vision for visual detection and localization need
33 more advances of this area essentially for illumination changes and accuracy in
34 feature extraction and matching with low complexity of computation needed for
35 mobile devices.

36 Solutions for this problem have been proposed by several authors. The main idea
37 is to generate panoramas from rotational motion of the device. The panorama is
38 created in a way that allowed users to annotate objects and to be shared with other
39 users visiting the same spot as annotations anchor points were redetected in newly
40 created panoramas by matching small image patches (Langlotz 2010, 2011). In
41 similar work, AR system which allows to users to create content is proposed in
42 Langlotz et al. (2012). To do this, authors propose two different approaches for
43 tracking the device position: in a small workspaces using natural feature-based
44 tracking of a known planar surface and in a large environment using GPS and
45 panorama-based vision tracking.

46 Our goal is to avoid the complexity of computer vision techniques and the
47 required accuracy for different tasks. We propose an efficient method of low
48 complexity for textual annotating of places using a new concept: circular string of
49 identifiers describing the 360° neighbour of the mobile device. The orientation, the
50 GPS localization are used in order to define the arc of the circular string corresponding
51 to what is seen. From the content of this arc, the annotation is displayed.

52 We implemented this method in our university campus for helping new students
53 to discover all building and places.

54 In Section 2 we describe the notion of circular string and how is used to infer the
55 seen places. The Sect. 3 explains the solution for taking into account the motion of
56 the device and the appearance of new places around the device. We describe the
57 obtained results in Sect. 4. Finally, we conclude giving some possible improvements
58 and future works.

59 2 Coding Seen Places Using Circular String of Identifiers

60 2.1 Basic Principle

61 For a given position O representing the mobile device we associate on the map a
62 circle (C) centred on O (see Fig. 1). Any place P_i surrounding the device that can
63 be seen by the camera (when is fixing around it) is projected as an arc on the circle
64 (C) labelled with an identifier (a character) which corresponds to a string
65 annotating P_i .

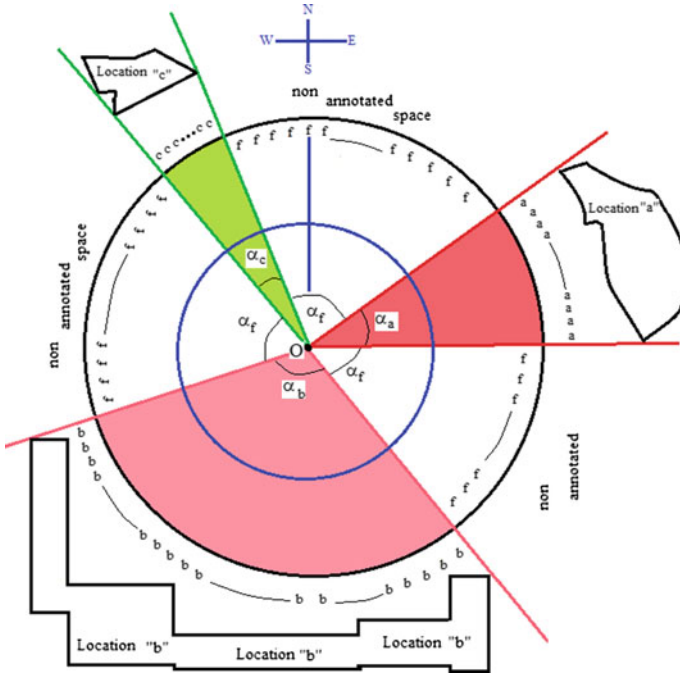


Fig. 1 The circular string of identifiers for a given location O composed here by the characters f (free space), a (location “a”), b (location “b”), c (location “c”)

66 The circle (C) is represented by a circular string “**SeenLocations**” of 360
 67 identifiers. The identifier attached to each arc is duplicated n times, where n is the
 68 length of the arc. The starting identifier in **SeenLocations** coincides with the north
 69 direction and that whatever the radius of the drawn circle, the obtained string is the
 70 same because the lengths of the different arcs are the same (see the blue circle in
 71 Fig. 1). In the example of Fig. 1, the content of this string is:

72 **SeenLocations** = “ $aaaa...aaaa \ fff...fff \ bbbb...bbb \ fff...ff \ ccc...cccc \ fff...ffff$ ”,
 73 where f designates the no-annotated free space.

74 Note here that the dictionary of places (identifier, string for annotation) is build
 75 off-line for a given area.

76 2.2 Determining the Seen Places from

77 Once the azimuth angle β and the GPS coordinates have been obtained, the optical
 78 axis of the camera and the position of O are located on the map (see Fig. 2). The
 79 visible space is determined based on the angle β and the field of view $(-\alpha, +\alpha)$
 80 of the camera. This allows defining a substring on the circular string which points

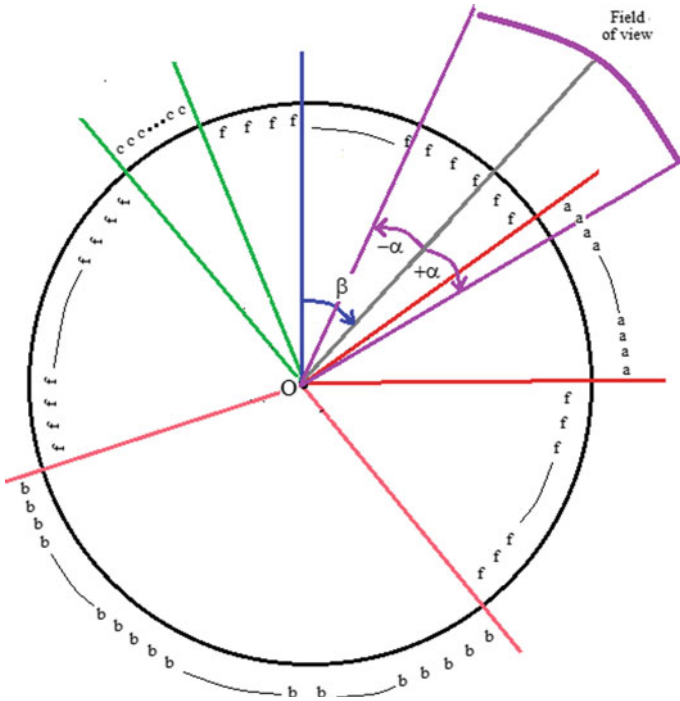


Fig. 2 Substring inference from azimuth angle β and field of view

81 either completely or partially to seen locations. On the string, the indexes of the
 82 substring are $\beta - \alpha$, $\beta + \alpha$. The displayed text will be the strings corresponding to
 83 each identifier of the arc. Depending on the content of the substring, the string is
 84 displayed as such or preceded by “part of”.

85 3 Updating the Circular String for Moving Mobile Device

86 3.1 Updating Without Integrating New Places

87 When the mobile device moves, the associated circular string must be modified
 88 because any selected substring of identifiers will not refer necessarily to the correct
 89 place. We assume that the device is moving from O to O' (see Fig. 3), the arc (mn)
 90 associated to the place P in the circle (C) corresponds to the arc $(m'n')$ on the circle
 91 (C') whose length and position may be different.

92 Let A, B be two of corners delimiting the place P . Let m'', n'' be new located
 93 points on (C') at the same positions as m, n on (C) . The arc $m''m'$ defines the angle
 94 α_m is equal to $\widehat{OAO'}$ which is computed using the distances $OO', O'A, OA$. In the

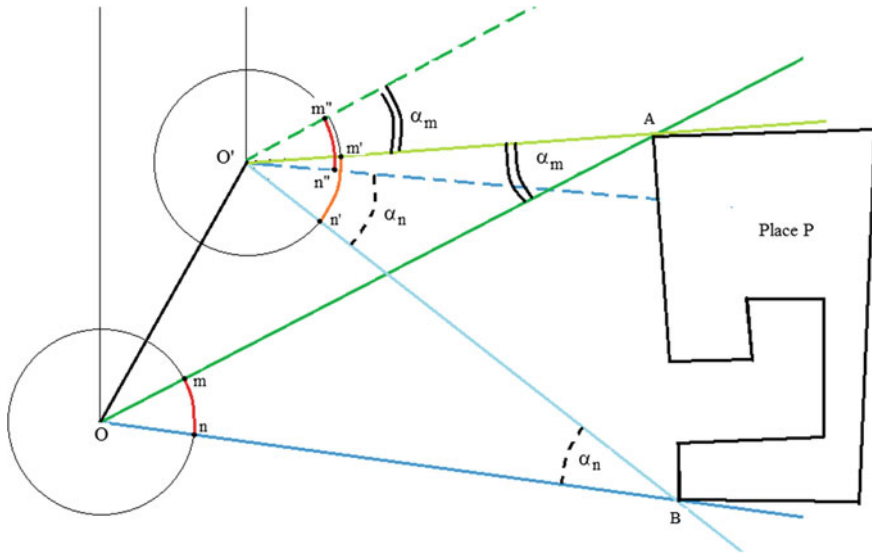


Fig. 3 Variation of the arc of view related to the mobile device motion

95 same way, the arc $n''n'$ defines angle α_n equal to angle $\widehat{O'BO'}$ which is computed
 96 using the distances $OO', O'B, OB$. Consequently, the positions of O, A, B, O'
 97 allows computing angles α_m, α_n and then the positions of m', n' corresponding to
 98 the seen place on the circular string.

99 The GPS coordinates of O, O' are communicated by the device (client) to the
 100 server, while those of corners A, B are assumed known (extracted and stored
 101 off-line from *OpenStreetMap* and *Leaflet* library). The content of the circular string
 102 is updated for each motion of the camera. The next algorithm describes the steps to
 103 be performed.

104 3.2 The Algorithm

105 Let P_0, P_1, \dots, P_{k-1} be k places of a given area. $P_i(A_{i,0}, A_{i,1}, \dots, A_{i,n_i}, id, Str_i)$
 106 refers to the place P_i where $A_{i,0}, A_{i,1}, \dots, A_{i,n_i}$ define the corners of the convex
 107 hull encompassing it, Str_i is the annotation and id the referred identifier on the
 108 circular string. The algorithm 1 gives more details how the content of the circular
 109 string is updated (see Fig. 4).
 110

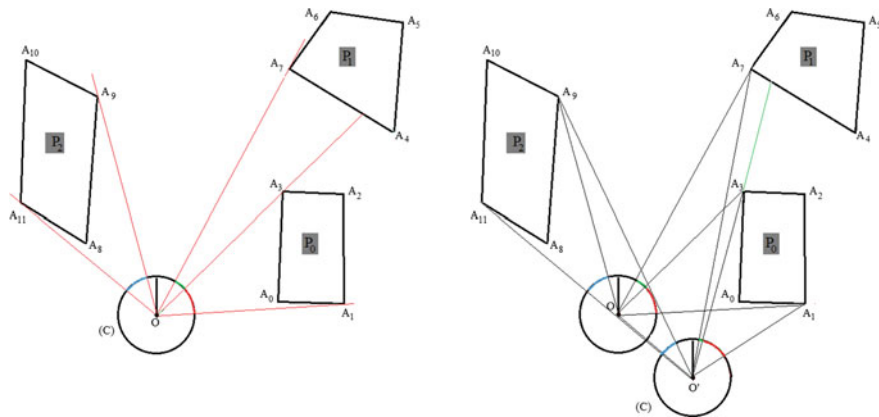


Fig. 4 Applying the algorithm 1 on the area of three places. (Left) initializing the circular string, (right) updating it

Algorithm 1.

Begin

#Initializing the circular string $SeenLocations[i] = 'f'$ for $i=0$ to 359.

For($i=0$ to 359) $SeenLocations[i] = 'f'$

For each place $P_i(A_{i,0}, A_{i,1}, \dots, A_{i,n_i}, id, Str_i)$

{ **For each** corner $A_{i,j}$

{

- Compute the associated arc mn on the circle (C) using the rays $OA_{i,j}$.
- **If** $\exists l / OA_{i,l} \cap P_j \neq \emptyset$ such that P_j is near to (C) than P_i ,
Then consider the ray that links O to the corner of P_j and passes throughout P_i . **EndIf**

}

Let (pos1, pos2) be the positions of the arc mn on (C) .

For($i=pos1$ to $pos2$) $SeenLocations[i] = 'id'$

}

#Updating the circular string

#Let O' be the new position of the camera

For each arc mn of (C) associated to $P_i(A_{i,0}, A_{i,1}, \dots, A_{i,n_i}, id, Str_i)$

{ #Let $A_{i,l}, A_{i,r}$ be the corners of projected on the arc mn on (C)

- Compute the new position m' on (C) located so that $mm' = \widehat{OA_{i,l}O'}$
- Compute the new position n' on (C) located so that $nn' = \widehat{OA_{i,r}O'}$
- **If** $A_{i,l}$ or $A_{i,r}$ is not a corner of P_i

Then evaluate the new projection on (C) using the nearest place P_j . **EndIf**

Let (pos1, pos2) be the positions of the arc $m'n'$ on (C) .

For($i=pos1$ to $pos2$) $SeenLocations[i] = 'id'$

}

End.

3.3 Updating with Integration of New Places

When the device is in the same area, it may be sufficient to update the circular string as explained in subsection 3.1. However, when it moves towards another area or when new convex hulls' segments appear, the content of the circular string must be changed by inserting the visible places. Depending on the convex hull encompassing the place P , and on the position of the device's centre O , the corner points of P which will be used are determined depending on the position of O related to convex hull (see Fig. 5). Let $C_{i,4}$, $C_{i,0}$, $C_{i,1}$ be three successive corner points of the convex hull, and let (D_0) , (D_1) be the line defined respectively by $(C_{i,4}, C_{i,0})$, $(C_{i,0}, C_{i,1})$. If the centre O appertains the area delimited only by (D_0) , (D_1) and $C_{i,0}$ (see Fig. 5), the corners seen are $C_{i,4}$, $C_{i,0}$, $C_{i,1}$. This reasoning is applied for the current position of the centre O and the area where is located. As example, if the device is in the area delimited by (D_1) , (D_4) , the corners $C_{i,3}$, $C_{i,4}$, $C_{i,0}$, $C_{i,1}$ are visible.

Our method allow then to infer the circular string giving the new GPS coordinates of the mobile device and the known GPS coordinates of the corner points of all places of the area considered.

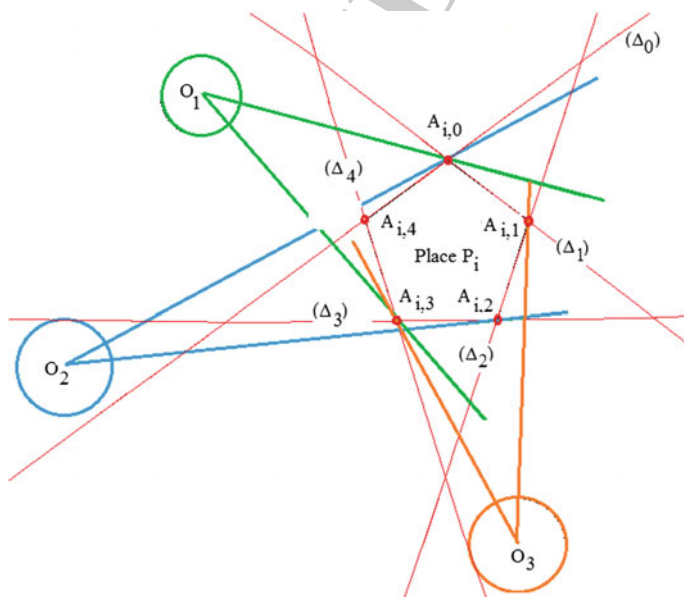


Fig. 5 Used corners in the computation of the arc of view



Fig. 6 The circular string of identifiers on the map for a selected reference point

4 Experiments and Results

We implemented the proposed method on mobile device under the system android. Some reference locations have been chosen in our university and the data related to the circular have been obtained manually and used as input data to the application. We show in Fig. 6 the map (one area of our university) obtained using the library Leaflet and *OpenStreetMap*. A reference point is chosen, the circle is drawn and the different arcs are determined. For each arc is associated an identifier (character) and the circular string is made using the lengths of these arcs. For each identifier is associated a text for annotation.

The selected views are illustrated by Fig. 7 where we can see the images annotated with correct text.

In order to avoid the instability of GPS measure which leads to instability of the text, we considered intervals of values instead of once value.

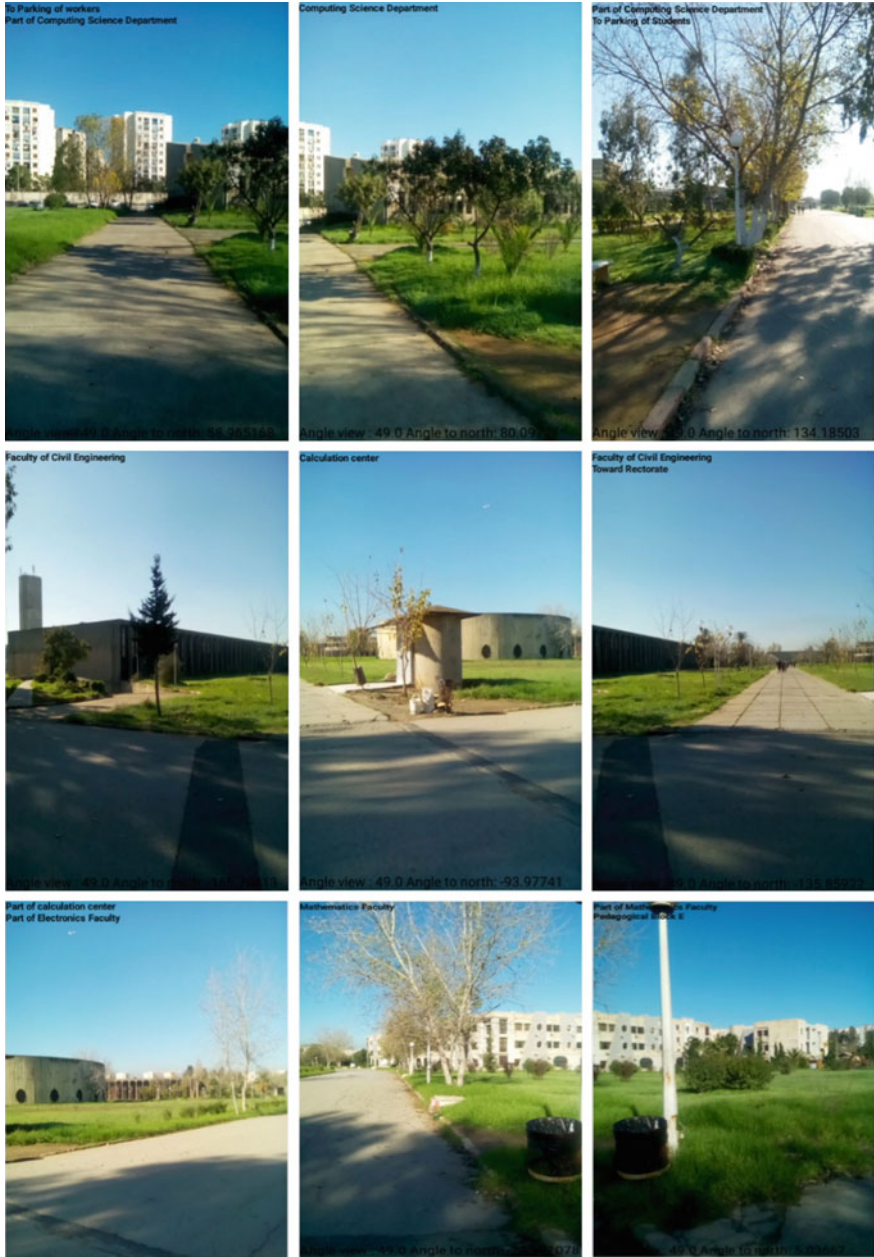


Fig. 7 The annotated locations

5 Conclusion

We proposed a geometry based method for outdoor places annotation. Our method has been implemented and applied to our university campus and served for new students to discover using their mobile devices the different places without help.

The notion of circular string of identifiers introduced for coding the seen places has gave more efficiency to the application in the sense that it is easy to retrieve the places seen entirely or partially. This circular string is also updated for any motion of the device based on stored information of the area. Our next tasks are to deploy this method for tourism areas adding some text explaining the historical monuments and for annotating streets, buildings in any city of the world.

Acknowledgements My acknowledgements to my Master students (MIV) I. MERABTENE, N. BENSALDANE, M. OUSMER, K. MESSIPSA, M. CHETITAH who have implemented the first part of this work that I gave as project during classes on creative workshop.

References

- Eaddy, M., Blasko, G., Babcock, J., & Feiner S. (2004). My own private kiosk: privacy-preserving public displays. In: *Proceedings of the international symposium on wearable computers*, pp. 132–135.
- Feiner, S., MacIntyre, B., Hollerer, T., & Webster, A. (1997). A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In: *Proceedings of the international symposium on wearable computers*, pp. 74–81.
- Langlotz, T., Degendorfer, C., Mulloni, A., Schall, G., Reitmayr, G., & Schmalstieg, D. (2011). Mobile Augmented Reality, Robust detection and tracking of annotations for outdoor augmented reality browsing. *Computers & Graphics*, 35(2011), 831–840.
- Langlotz, T., Mooslechner, S., Zollmann, S., Degendorfer, C., Reitmayr, G., & Schmalstieg, D. (2012). Sketching up the world: In situ authoring for mobile Augmented Reality. *Pers Ubiquity Computing*, 16(2012), 623–630.
- Langlotz T., Wagner, D., Mulloni, A., & Schmalstieg, D. (2010). Online creation of panoramic augmented reality annotations on mobile phones. *IEEE pervasive computing*.
- Liberati, N. (2016). Augmented reality and ubiquitous computing: The hidden potentialities of augmented reality. *AI & SOCIETY*, 31(2016), 17–28.
- Ryu, H. S., & Park, H. (2016). A system for supporting paper based augmented reality. *Multimedia Tools & Applications*, 75(2016), 3375–3390.
- Schmalstieg, D., & Wagner, D. (2007). Experiences with handheld augmented reality. *Proceedings of the international symposium on mixed and augmented reality, 2007*, 3–18.
- Wither, J., DiVerdi, S., & Hollerer, T. (2009). Annotation in outdoor augmented reality. *Computers & Graphics*, 33(6), 679–689.

Author Query Form

Book ID : 436951_1_En

Chapter No : 24



Please ensure you fill out your response to the queries raised below and return this form along with your corrections.

Dear Author,

During the process of typesetting your chapter, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query Refs.	Details Required	Author's Response
AQ1	Please confirm if the inserted city name is correct. Amend if necessary.	
AQ2	Reference Feiner 1997 is present in the list and not cited in the text. Please provide the citation.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

<i>Instruction to printer</i>	<i>Textual mark</i>	<i>Marginal mark</i>
Leave unchanged	... under matter to remain	Ⓟ
Insert in text the matter indicated in the margin	∧	New matter followed by ∧ or ∧ [Ⓢ]
Delete	/ through single character, rule or underline or ┌───┐ through all characters to be deleted	Ⓞ or Ⓞ [Ⓢ]
Substitute character or substitute part of one or more word(s)	/ through letter or ┌───┐ through characters	new character / or new characters /
Change to italics	— under matter to be changed	↙
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	≡ under matter to be changed	≡
Change to bold type	~ under matter to be changed	~
Change to bold italic	≈ under matter to be changed	≈
Change to lower case	Encircle matter to be changed	≡
Change italic to upright type	(As above)	⊕
Change bold to non-bold type	(As above)	⊖
Insert 'superior' character	/ through character or ∧ where required	Υ or Υ under character e.g. Υ or Υ
Insert 'inferior' character	(As above)	∧ over character e.g. ∧
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	ʹ or ʸ and/or ʹ or ʸ
Insert double quotation marks	(As above)	“ or ” and/or ” or ”
Insert hyphen	(As above)	⊞
Start new paragraph	┌	┌
No new paragraph	┐	┐
Transpose	└┘	└┘
Close up	linking ○ characters	○
Insert or substitute space between characters or words	/ through character or ∧ where required	Υ
Reduce space between characters or words		↑