(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 August 2013 (22.08.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/122536 A1

(51) International Patent Classification:
G06T 19/00 (2011.01) *G06K 9/00* (2006.01)

(21) International Application Number:
PCT/SE2013/050121

(22) International Filing Date:
13 February 2013 (13.02.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1250117-7 14 February 2012 (14.02.2012) SE

(71) Applicant: **13TH LAB** [SE/SE]; Johannesgatan 8, S-111 38 Stockholm (SE).

(72) Inventors: **LINDE, Oskar**; Barks Väg 16, S-170 73 Solna (SE). **BOND, Anders**; Torpvägen 14, S-175 50 Järfälla (SE). **IVMARK, Petter**; Skanörvägen 16, S-121 52 Johanneshov (SE).

(74) Agent: **KRANSELL & WENNBORG KB**; P.O. Box 27834, S-115 93 Stockholm (SE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: ADDING AUGMENTATION DATA TO A 3D PERSPECTIVE VIEW

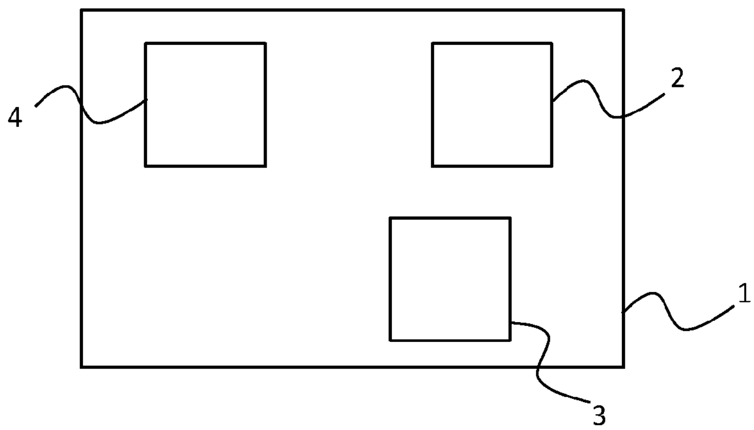

FIG. 1

(57) **Abstract**: The present invention relates to a method for adding augmentation data to a 3D perspective view of an object in front of a camera device (2) of a mobile camera arrangement (1), the method comprises the steps of: determining a perspective value of the camera device of the mobile camera arrangement; receiving frames of image data of the object from the camera device of the mobile camera arrangement; determining a camera position of the camera device for a frame of the received frames; determining a transform for the frame of the received frames, based on the camera position; and in a browser (4) displaying augmentation data in a 3D perspective view of the frame of the received frames, wherein the augmentation data is resized by the transform in relation to the object.

WO 2013/122536 A1

Adding augmentation data to a 3D perspective view

FIELD OF INVENTION

The present invention relates generally to camera devices and more particularly to a method for adding augmentation data to a 3D perspective view of an object in front of a camera device of a mobile camera arrangement.

BACKGROUND

Most mobile devices today are equipped with a camera allowing real time viewing of events in front of the camera on a display of the mobile device. Viewing of events on the display from a 3D angle is desirable, and adding of augmentation data in a correct 3D perspective to the picture displayed on the display device is also desirable.

SUMMARY OF THE INVENTION

In case augmentation data is to be displayed in a 3D perspective on a display of a mobile communication arrangement in relation to an object in front of a camera device in the mobile communication arrangement, it is desirable to provide the augmentation data in relation to movement of the camera device relative the object, which makes programming of the augmentation data complex and resource demanding.

An object of the present invention is to provide a method of adding augmentation data in a 3D perspective to a view in relation to an object in front of a camera device of a mobile communication arrangement, without a developer having to consider movement of the camera device relative the object when programming the augmentation data.

This object is according to the present invention attained by a method as defined by the appended claims.

At an insight of the inventors a method is provided allowing e.g. standard HTML and JavaScript code and content related to a certain event or a physical object in front of a camera device to be correctly rendered on top of or relative to that object as the camera device is being moved around by a user, without the developer of the HTML content having to understand or adapt the rendering of the content himself as its perspective on the screen is shifted to reflect the movement of the camera device relative to the object.

By providing a method for adding augmentation data to a 3D perspective view of an object in front of a camera device of a mobile camera arrangement, the method comprises the steps of: determining a perspective value of the camera device of the mobile camera arrangement; receiving frames of image data of the object from the camera device of the mobile camera arrangement; determining a camera position of the camera device for a frame of the received frames; determining a transform for the frame of the received frames, based on the camera position; and in a browser displaying augmentation data in a 3D perspective view of the frame of the received frames, wherein the augmentation data is resized by the transform in relation to the object it is possible to in e.g. HTML add augmentation data in a correct 3D perspective to a view of an object in front of a camera device of a mobile camera arrangement without knowledge of movement of the camera device relative the object, thus allowing a developer to render traditional HTML and JavaScript content on top of reality.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will become more fully understood from the detailed description of embodiments given below and the accompanying figures, which are given by way of illustration only, and thus, are not limitative of the present invention, wherein:

Fig. 1 is a schematic illustration of a mobile camera arrangement according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, for purpose of explanation and not limitation, specific details are set forth, such as particular techniques and applications in order to provide a thorough understanding of the present invention. However, it will be apparent for a person skilled in the art that the present invention may be practiced in other embodiments that depart from these specific details. In other instances, detailed description of well-known methods and apparatuses are omitted so as not to obscure the description of the present invention with unnecessary details.

An embodiment of a method for adding augmentation data to a 3D perspective view of an object in front of a camera device of a mobile camera arrangement will now be described with reference to Fig. 1.

The method is exemplified for iOS and WebKit, and comprises the following steps.

Firstly, a perspective value of the camera device of the mobile camera arrangement is determined, preferably based on at least the focal length of the camera in video pixels, the video width in pixels and the width of the displayed video in screen pixels. The perspective value is calculated by first dividing the focal length of the camera measured in video pixels by the video width in video pixels, thereafter multiplying the result by the displayed video stream width, measured along the same axis, in screen pixels. The perspective value is thereafter preferably set to a perspective node of an HTML page e.g. by calling a JavaScript method. The perspective node is preferably an HTML div named e.g. viper.world node and the perspective value is given as a parameter to the CSS attribute “-webkit-perspective”. The cameras optical centre is re-calculated to screen pixels in the web view and given as parameter to the CSS attribute “-webkit-perspective_origin” of the perspective node. Also a scene node is preferably provided in an HTML div named e.g. viper.scene_node. The scene node is preferably given the CSS property "-webkit-transform-style: preserve-3d”, wherein any content inside it is transformed with a scene transform. The perspective node is preferably directly within the <body> tag of the HTML document, and the scene node is preferably located within the perspective node. An advantage of providing the world and scene as two separate HTML divs, compared to both being in the same div is that it makes it easier for a developer to work with.

The perspective value is here described as a calculation in pixels. This can of course be calculated in other forms such as in mm.

Frames of image data of the object are received from the camera device of the mobile camera arrangement. A camera position of the camera device is determined for a frame of the received frames, preferably for each frame of the received frames. The camera position relative to a 3D scene in the real world, or a subset of the real world, such as a surface or object, is preferably estimated from SLAM (Simultaneous Localization And Mapping), image or surface recognition, object recognition or other localization techniques, which are suitable for mobile device processing. The camera position calculation is preferably performed in the native layer of the application using e.g. C/C++ or Objective-C.

A transform for the current frame of the received frames is thereafter determined based on the estimated camera position. The transform is preferably also performed in the native layer, and comprises identification of how the object is positioned and moves relative to a 3D coordinate system, i.e. being a coordinate system transform. The transform is preferably given as a parameter to CSS attribute "-webkit-transform: matrix3d(…)" of the scene node. The transform is preferably determined for each frame of the received frames.

Setting of CSS attributes of nodes in the HTML page from the native part of the application are e.g. done through an evaluation of a JavaScript method such as a native-to-JavaScript call (stringByEvaluatingJavaScriptFromString).

Augmentation data, in this case augmentation HTML and JavaScript data, is in a browser displayed to a 3D perspective view of the frame of the received frames, wherein the augmentation data is resized by the transform in relation to the object. Images used to initiate a 3D perspective view may specify actual (real world) dimensions. In such a case the scale of a scene is calculated to correctly match reality.

When an HTML view's resolution in pixels is specified an HTML page scale is used, which preferably is taken into account when calculating a scene node 3D transform matrix. A default value of the scale may be 1000, meaning that an image sized 0.5 x 0.5 meters in reality will be completely covered by a content sized 500 x 500 pixels.

A method of detecting and tracking an event in front of the camera device is preferably utilized in combination with the method of adding augmentation data as described above. A possibility is provided for a JavaScript program to be able to receive notifications of specific events that happen in front of the camera of a mobile device, such as a specific picture or object being held up in front of the camera and thus in real-time adapt the content of the web page to what the camera device is pointed at, or based on certain events that happen in front of it. This is a great improvement in the possibility of customization of the browser content and the ability to make that content dynamic over the traditional methods of using GPS or Wi-Fi location to adapt the web page's content to the camera device's surroundings. To do such things has previously required either writing, downloading and installing a dedicated application program (e.g. an iOS or Android application) for each use case that may be conceived (e.g. one application for recognizing articles in news papers, another application for recognizing museum art), or creating static content in bespoke/proprietary content formats that need to be published to specific content hosts and cannot be updated in real time by a developer, giving the developer no way to run/execute code client side as the application is used and thus dynamically react to events as can be done with a client side application and is also the case with a normal web page on the Internet. Using the default browser instead, developers can create normal web pages, running client side JavaScript code, that they host themselves and a user can simply load a new page (and thus the developer's JavaScript program) from different places on the Internet for different use cases just like on the rest of the Internet.

The method offers a solution for JavaScript programs, running in a built-in browser of a mobile camera arrangement, to be notified of certain events that happen in front of a camera device thereof and for the JavaScript program embedded in a web page loaded in the built-in browser to react to these events in real time. An example is a web page for a museum that adapts and presents information about a certain work of art automatically as the device's camera is pointed towards it.

The certain events can e.g. be generic, such as "faces", which would then notify the JavaScript program that a face was found in front of the camera, preferably along with other features such as the orientation, scale and position of that face in the image or in 3D space. Certain events could e.g. also be known images or objects found in front of the camera that were provided by the JavaScript program at run-time, when the page was loaded in the browser, such as a set of known images, e.g. a specific set of paintings, that the JavaScript program of the web page would like to get notified about as they appear in front of the camera device.

Most mobile devices today are equipped with a built-in mobile browser that contains a JavaScript run-time environment. This run-time environment allows the mobile device to download web pages that execute program code directly in the browser, without having to install an application program in the mobile device. This is very advantageous in many ways as the barrier to using the program is much less than when having to firstly install a dedicated software application in the mobile device for each possible use case. It is also very advantageous for developers as it utilizes standardized programming languages such as HTML and JavaScript that are less complicated to use then lower level languages such as C/C++/Objective-C/Java and do not require each developer to write and distribute their own native application for a potentially small use case.

Just like other run-time environments (e.g. Objective-C on iOS and Java on Android), the JavaScript run-time environment often has certain APIs (Application Program Interfaces) offered by the device manufacturer to be able to communicate with hardware and sensors in the mobile device. An example of this is that a JavaScript program can often ask the mobile device about the mobile device's GPS position via a JavaScript API exposed by the manufacturer.

These APIs are however often limited to what the manufacturer has chosen to offer access to and not easily extensible to other use cases than those pre-defined by the manufacturer.

Using such manufacturer provided APIs, developers try to make their application programs more adapted to the users environment (e.g. presenting content based on the location of the mobile device). Today, access to such input is however very limited.

The mobile camera arrangement 1 is typically a smart phone having a built-in camera device 2. The mobile camera arrangement is typically configured to allow a custom application program 3 to be installed therein, allowing communication through an API having a set of native instructions for communication to other parts of the mobile camera device. The mobile camera arrangement 1 is exemplified by an iPhone, running iOS.

The custom application program 3 is installed in the iPhone creating functions for capturing events in front of the camera of the iPhone and exposing these functions via a set of APIs. The custom application program 3 is when run configured to embed the iOS operating system's built-in browser 4 as a so called web view and allow a standard webpage and JavaScript program therein to use these additionally exposed APIs in order to request and be notified of events in front of the camera of the iPhone.

The method for allowing a JavaScript program in a web page to be notified comprises the steps of:

from a JavaScript program running in the system's built-in browser requesting notifications of a certain event happening in front of the camera device of the mobile camera arrangement;

analysing data from the camera device of the mobile camera arrangement;

detecting the certain event from the analysed data; and

notifying the JavaScript program in the built-in browser of the detected certain event.

The certain event is preferably requested from the webpage running in the built-in browser embedded by the custom application program. The step of analyzing data and the step of detecting the certain event are preferably executed by the custom application running in the mobile camera arrangement.

The data from the camera device can e.g. be one or more camera images, or a stream of camera images. In case the certain event is detected in more than one image, or in multiple positions in one image, the JavaScript program is preferably notified of each such detected position. The camera images are preferably collected from the camera device. After the camera images have been collected, the images are analyzed in a known way by image, object, scene or other pattern or 2D or 3D recognition searching for a match to the certain event. The data from the camera device can also e.g. be image descriptors or a video stream.

The step of requesting a certain event preferably comprises sending the request for the certain event through the custom API exposed by the custom application program. Installation of the custom application program thus creates a set of APIs that the webpage and JavaScript program can use to register for certain events for which it would like to get notified about. An example could be to register to be notified of faces found in the cameras view, which could be provided by the following JavaScript code: registerForCameraEvent("face", myFaceEventCallback);

While on many systems, such as in this case the iOS, the built-in browser in a mobile camera arrangement can only communicate to and from the native application layer of the mobile camera arrangement using the pre-defined APIs provided by the manufacturer himself (for example a camera API or a location/GPS API), the present invention utilizes a way of circumventing this restriction to allow the custom application program to communicate with, receive from, and send data to, the built-in browser provided by the manufacturer.

An example of how this can be done in iOS (there are also other ways this could be done depending on the manufacturer's browser implementation) is described below:

To send data from the JavaScript program to the native layer, the application program uses a previously defined custom URL scheme with a specific format that the native application is aware of (in this example using window.location or a hidden iframe to initiate the request) to start monitoring a certain event.

On the native side the custom application program can intercept this request in a callback before it is loaded and processed by the built-in browser. The URL request is examined, and if it matches a specific format for a request (for example fakerequest://command?param1=1&param2=2), the URL request is interrupted and it is parsed to get the command and data for the event that the JavaScript program is requesting. The request from the web page is then terminated, causing the browser not to load a new page. If it's not a match, the URL request goes through uninterrupted.

An alternative way to communicate from the JavaScript program running in the embedded browser view, to the native application is to register a custom URI scheme such as e.g. pointcloud:// on the device using the info.plist file and then perform calls to the native part of the program using the format pointcloud:// command?param1=1&param2=2

To communicate from the native application to the JavaScript program running in the built-in browser the invention uses a way to execute a JavaScript command from the native application. The native application program can use a method included in the built-in browser API (such as UIWebView:stringByEvaluatingJavaScriptFromString in iOS). The application program will then send a JavaScript command as a string, which will be executed by the built-in browser and a result returned to the JavaScript program in the browser.

Another alternative way to achieve the same functionality is to let the native part of the application (in this case the Objective-C part of the application) set up a web server on e.g. the local host address 127.0.0.1 of the device and let the JavaScript part of the application (in this case the embedded web view) communicate with the native part of the application using HTTP requests or web sockets to this web server on the same device.

The method steps of the present invention are preferably performed by computer program code of a computer program for the mobile camera arrangement. The computer program can preferably be stored on a computer readable means forming a computer program product.

A second embodiment of the present invention will now be described. This second embodiment of the present invention is identical to the first embodiment described above apart from the following.

In the second embodiment the method for allowing JavaScript in a mobile web browser is adapted for the mobile camera arrangement instead being an Android phone. In order for the JavaScript in the webpage of the built-in browser to be able to be notified of certain events in front of the camera the native application has created a set of Java functions that unlike iOS can be exposed directly to the

JavaScript program in the browser, thus letting the JavaScript program directly do pre-defined native function calls from the browser in order to request notification of certain events.

To communicate from the native application to the JavaScript application in the built-in browser, the custom application program on Android can also utilize the Java-based APIs that unlike iOS can be exposed directly to the JavaScript program in the browser, allowing the native application to do a so called call-back to the JavaScript program when an event occurs in front of the camera.

It will be obvious that the present invention may be varied in a plurality of ways. Such variations are not to be regarded as departure from the scope of the present invention as defined by the appended claims. All such variations as would be obvious for a person skilled in the art are intended to be included within the scope of the present invention as defined by the appended claims.

CLAIMS

1. A method for adding augmentation data to a 3D perspective view of an object in front of a camera device (2) of a mobile camera arrangement (1), **characterized in that** said method comprises the steps of:

determining a perspective value of said camera device of said mobile camera arrangement;

receiving frames of image data of said object from said camera device of said mobile camera arrangement;

determining a camera position of the camera device for a frame of the received frames;

determining a transform for said frame of the received frames, based on said camera position; and

in a browser (4) displaying augmentation data in a 3D perspective view of said frame of the received frames, wherein said augmentation data is resized by said transform in relation to said object.

2. The method according to claim 1, wherein said step of determining a camera position is performed in a native layer, preferably using low level high performance programming languages such as C/C++/Objective-C/Java.

3. The method according to claim 2, wherein said step of determining a transform is performed in said native layer, and said step comprises identification of how said object moves relative a 3D coordinate system.

4. The method according to claim 3, comprising a step setting the determined transform to a CSS parameter.

5. The method according to any of claims 1-4, wherein said perspective value is based on at least the focal length of the camera device, the video width of the camera device, and the displayed video stream width.

6. The method according to claim 5, wherein said perspective value is calculated by first dividing said focal length of the camera device measured in video pixels by said video width in video pixels, thereafter multiplying the result by said displayed video stream width, measured along the same axis, in screen pixels.

7. The method according to any of claims 1-6, wherein said perspective value is set to a perspective node and said transform is set to a scene node.

8. The method according to claim 7, wherein the cameras optical centre is re-calculated to screen pixels in a web view and given as a CSS parameter of said perspective node

9. The method according to any of claims 1-8, wherein said perspective value is set to a perspective node by calling a JavaScript program running in a browser in said mobile camera arrangement.

10. The method according to any of claims 1-9, wherein said camera position is determined for each frame of the received frames.

11. The method according to any of claims 1-10, wherein said camera position is determined based on an estimation by SLAM, image recognition, surface recognition, object recognition, or other localization techniques.

12. The method according to claim 9, comprising the step of:

from said JavaScript program running in said browser requesting notifications of a certain event happening in front of said camera device.

13. The method according to claim 12, comprising the steps of:

analysing said frames of image data;

detecting said certain event from the analysed frames of image data; and

notifying said JavaScript program running in said default browser of the detected certain event.

14. A computer program for a mobile camera arrangement, the computer program comprising computer program code which performs method steps of the method according to any of the previous claims

15. A computer program product comprising a computer program according to claim 14 and a computer readable means on which the computer program is stored.

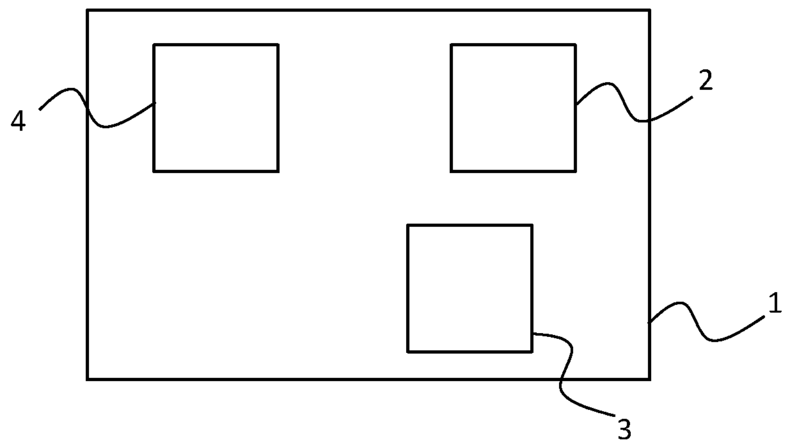

FIG. 1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE2013/050121

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06K, G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE, DK, FI, NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, PAJ, WPI data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 20100257252 A1 (DOUGHERTY MICHAEL A ET AL), 7 October 2010 (2010-10-07); abstract; paragraphs [0006]-[0008], [0028]-[0030], [0037]-[0057]; figure 6	1-11, 14-15
Y		12-13
	--	
Y	US 20090285444 A1 (EROL BERNA ET AL), 19 November 2009 (2009-11-19); abstract; paragraphs [0001]-[0010], [0040]-[0046], [0064]-[0065], [0090]-[0093]; figures 1,6-7	12-13
A		1-11, 14-15
	--	

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17-05-2013	20-05-2013

Name and mailing address of the ISA/SE	Authorized officer
Patent- och registreringsverket Box 5055 S-102 42 STOCKHOLM Facsimile No. + 46 8 666 02 86	Jimmie Femzén Telephone No. + 46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE2013/050121

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"The Argon AR Web Browser and standards-based AR application environment", Blair MacIntyre; Alex Hill; Hafez Rouzati; Maribeth Gandy; Brian Davidson, 2011-10-26 Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium doi:10.1109/ISMAR.2011.6092371 ISBN 978-1-4577-2183-0 ; ISBN 1-4577-2183-X; whole document --	1-15
A	US 20110164163 A1 (BILBREY BRETT ET AL), 7 July 2011 (2011-07-07); abstract; paragraphs [0002]-[0004], [0014]-[0029]; figures 1A-1D --	1-15
A	US 20090102859 A1 (ATHSANI ATHELLINA R AHMAD ET AL), 23 April 2009 (2009-04-23); abstract; paragraphs [0001]-[0011] -- --------	1-15

Continuation of: second sheet

International Patent Classification (IPC)

G06T 19/00 (2011.01)
G06K 9/00 (2006.01)

US	20100257252 A1	07/10/2010	NONE			
US	20090285444 A1	19/11/2009	CN	101582083	A	18/11/2009
			JP	2009295153	A	17/12/2009
			US	8385589	B2	26/02/2013
US	20110164163 A1	07/07/2011	US	8400548	B2	19/03/2013
US	20090102859 A1	23/04/2009	US	20130057585	A1	07/03/2013
			US	20130044132	A1	21/02/2013
			US	20120244907	A1	27/09/2012
			US	8275414	B1	25/09/2012
			US	8180396	B2	15/05/2012