CONTRIBUTED
P A P E R

Next-Generation Augmented Reality Browsers: Rich, Seamless, and Adaptive

This paper discusses the challenges and varying research approaches to building augmented reality browsers to discover and view content related to physical objects around a mobile device.

By Tobias Langlotz, *Member IEEE*, Thanh Nguyen, *Student Member IEEE*, Dieter Schmalstieg, *Member IEEE*, and Raphael Grasset, *Member IEEE*

ABSTRACT | As low-level hardware will soon allow us to visualize virtual content anywhere in the real world, managing it in a more structured manner still needs to be addressed. Augmented reality (AR) browser technology is the gateway to such structured software platform and an anywhere AR user experience. AR browsers are the substitute of Web browsers in the real world, permitting overlay of interactive multimedia content on the physical world or objects they refer to. As the current generation allows us to barely see floating virtual items in the physical world, a tighter coupling with our reality has not yet been explored. This paper presents our recent effort to create rich, seamless, and adaptive AR browsers. We discuss major challenges in the area and present an agenda on future research directions for an everyday augmented world.

KEYWORDS | AR browser; augmented reality (AR); mobile devices

I. INTRODUCTION

Since the first steps in the 1990s, augmented reality (AR) has undergone a tremendous development. While AR experiences used to require carrying bulky custom hardware [1], today, a smartphone is sufficient. This accessibility has prompted the adoption of AR applications by the general public: AR is now used in marketing (e.g., augmenting the pages of a magazine), for games (virtual characters on your tabletop or on the street), or home shopping (e.g., placing virtual furniture in your living room).

Yet, one of the most successful types of AR application is the equivalent of a desktop or mobile Web browser for the physical world, generally referenced as *AR brower* (see Fig. 1). Historically, the term was proposed by SPRXmobile when presenting its *Layar AR browser*[1] before being adopted by academics and industries (Wikitude, Junaio, 13th Lab, etc.). This type of applications is today downloaded or preinstalled on more than 50 million smartphones, and with the emergence of low-cost head-mounted displays (e.g., Google Glass), we can expect mass integration of this kind of applications in our everyday life.

AR browsers can present information registered directly to places or artifacts in the real world on top of the live videostream, such as icons of restaurants (Fig. 2). This directness reduces cognitive effort and provides an advantage over conventional "location-based" interfaces such as maps or lists. For connected browsing, registered icons can be hyperlinked to additional content if more information is desired.

However, commercial AR browsers have not yet reached their full potential. Despite the fact that AR browsers already underwent several fundamental iterations in hardware and software changes, they can still be seen as being *poor* in terms of content, not visually *seamless*, and rather *static* [2].

Manuscript received July 15, 2013; accepted November 20, 2013. Date of current version January 20, 2014. This work was supported in part by the European Union under Project CultAR (FP7-ICT-2011-9 601139) and by the Christian Doppler Laboratory for Handheld Augmented Reality.

T. Langlotz is with the Institute of Computer Graphics and Vision, Graz University of Technology, Graz 8010, Austria and also with the Department of Information Science, University of Otago, Dunedin 9016, New Zealand (e-mail: langlotz@icg.tugraz.at).

T. Nguyen, **D. Schmalstieg**, and **R. Grasset** are with the Institute of Computer Graphics and Vision, Graz University of Technology, Graz 8010, Austria (e-mail: thanh@icg.tugraz.at; schmalstieg@tugraz.at; raphael@icg.tugraz.at).

Digital Object Identifier: 10.1109/JPROC.2013.2294255

[1]http://www.sprxmobile.com/we-launched-layar-worlds-first-augmented-reality-browser-for-mobile/.

Fig. 1. *Futuristic vision of rich, seamless, and adaptive next-generation AR browsers.*

Within this paper, we describe some of the identified issues of the current-generation AR browsers and their historical development (scholarly but also commercially). We elaborate on how we can make the AR browser *rich*, *seamless*, and *adaptive* to our real world (see Fig. 1).

We advocate that the next-generation of AR browsers should be *rich* by integrating a wide selection of digital media types in a meaningful way and large quantity, requiring an architecture and interface supporting various media types, while also emphasizing social media. We further argue that the AR browser should *seamlessly* integrate this digital media information into the physical world, requiring precise tracking on the one hand and view management techniques on the other hand. Finally, we emphasize the need for *adaptivity* of AR browsers to their current context, making it necessary to understand the user's environment. We illustrate some of these objectives with selected outcomes from our research, and discuss open problems and research directions.

This paper is organized as follows. Section II provides an overview of current-generation architecture of AR browsers and challenges for next-generation AR browsers. Sections III–V present some background and our own work on addressing these challenges. Section VI discusses insights from our work, before we conclude this paper in Section VII.

In summary, the contribution of this paper is the structured analysis of major research topics related to the AR browser and achievements from both the AR community as well as our own research.

Fig. 2. *Wikitude, the first commercial AR browser application as an example of the current generation of AR browsers, incorporating GPS-tagged content rendered using screen-aligned icons.*

II. AR BROWSER OVERVIEW

AR applications are complex multimedia systems aimed at visually blending digital information into the physical world. In the following, we give an overview of AR browser software architecture in terms of data structure, software components, and limitations.

A. Data Structure

The main data items of AR browsers are georeferenced or object-referenced point of interests (POIs). Georeferenced information comes in a variety of forms, such as textual or pictorial data, or more rarely in the form of video, audio, or 3-D media [2]. Information is usually geographically referenced using longitude/latitude data (WGS84), or attached to specific objects via specific visual fingerprints (e.g., image recognition, fiducial markers with 2-D barcodes, etc.).

The storage format is usually proprietary (custom XML-like databases), and only recently HTML has been used in a limited form. Similar to newsfeeds, content is frequently organized in thematic *channels*.

Many current AR browsers reuse the existing Global Positioning System (GPS)-tagged information from traditional web applications to increase the amount of accessible information. Typical examples are Twitter feeds, Flickr pictures, or Wikipedia articles. They also support information specifically authored for each AR browser, commonly done through desktop map-based authoring tools provided by the browser company.

B. Software Components

Similar to many mobile information systems, the general architecture of mobile AR browsers follows a client–server model (Fig. 3).

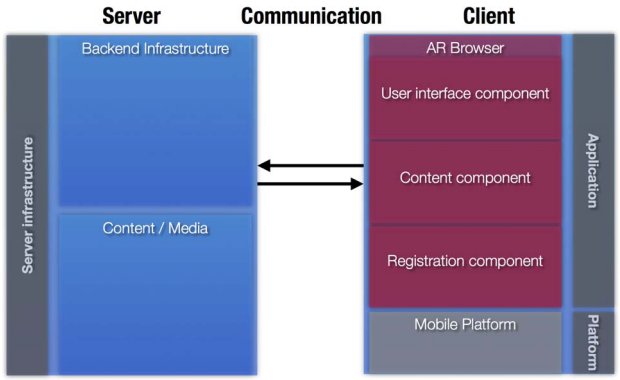


Fig. 3. ***Overview of the common architecture of an AR browser and its software components.***

The server is responsible for content storage and retrieval as queried by the client (e.g., provide all 3-D models in a 10-m radius around my location). The client is responsible for rendering the content and letting the user interact (browse). While the basic software architecture of mobile AR browsers is similar to 3-D games or simulations, AR requires some additional features.

First, the mobile device needs to determine its pose (position and orientation) in the physical world. It is required to identify surrounding information (what is around my position?), but also to transform any local information into the current user's view (what can I see from my current position?). These processes are usually done by a *registration component*. Commercial AR browsers usually employ built-in sensors—GPS for position and compass, accelerometers and gyroscopes for orientation—for this task.

This spatial information is then used by the *content component* to initiate streaming-relevant content from the server to the client. The *user interface component* is responsible for presenting the content. Combined visual presentation of real and virtual content can be achieved optically [1], but today the most convenient option is to add computer graphics to the digital camera feed of the built-in camera of a mobile device and present the result on the mobile device's screen.

C. Limitations

Despite the fact that AR browsers and their main architecture already have a long history in research, dating back to the original Touring Machine [1] over the Worldboard [3] to the real World Wide Web browser [4], the current generation of AR browsers can only be seen as an intermediate step on the way to fulfill the ultimate wishes already expressed in these earlier works. There are still significant shortcomings in terms of: 1) registration accuracy [5]; 2) insufficient quality and quantity of content [2], [6]; 3) an inflexible and proprietary software architecture of AR browsers [7]; and, finally, 4) poor usability of information presentation [8].

Consequently, future AR browsers require addressing the following research challenges.

- *Accurate and global registration*: While many techniques exist for computing the pose of mobile devices, most solutions do not support large and uncontrolled environment. This affects the seamless integration of virtual content, as users are burdened with registration problems.
- *Seamless registration*: Today's AR browsers rely on a combination of registration methods, such as for outdoor environments (GPS) or for object-based registration (feature matching, fiducial markers). Switching between these methods disrupts the user experience and should be made seamless in future AR browsers (for example, when moving from outdoors to indoors).
- *Content density*: AR browsers rely on situated media, but the distribution of situated media varies significantly. While city centers may be cluttered with POIs, rural areas may lack any interesting content. Ideally, we would like to have dense content coverage everywhere, with proper automated filtering and selection tools to manage clutter while browsing AR browsers.
- *Rich content*: While rich media are abundant in web applications, databases for AR browsers are heavily reliant on textual content and lacking in other media types.
- *Seamless content integration*: The user's understanding of a visually augmented scene is dependent on the quality of the integration of virtual information in the physical world, both spatially and perceptually. Unfortunately, AR browsers are not providing a satisfactory experience in these respects yet.
- *Adaptivity*: AR browsers can be used in a large variety of physical places. As our real world is constantly changing around us, AR browsers must be able to adapt. For instance, the contrast of computer-generated items may need to vary significantly from a sunny day to a cloudy day or indoor environment.

While some of these challenges must be addressed by improving existing components, such as registration or user interface, other challenges require inclusion of novel components, such as adaptive view management. In the rest of this paper, we present our work toward these challenges.

III. REGISTRATION

A fundamental component of every AR application is the precise registration of virtual content within a real-world coordinate system. Commonly, registration is a two-step process, a localization step followed by a tracking step. The localization delivers a onetime accurate absolute pose,

while the tracking provides a continuous relative pose. Pose should be determined with six degrees of freedom (6DOFs) and tracked at 30 Hz or above, yielding an accuracy of a fraction of a degree in orientation and a few millimeters in position [5]. Satisfying these requirements in practice remains challenging, especially in outdoor environments.

A. Background and Challenges

Original research on AR browsing used expensive sensing equipment such as real-time kinematic GPS to obtain acceptable accuracy [9]. Today's AR browsers on smartphones only have consumer grade sensors at their disposal. Even when assisted by other sensors, such as WiFi or the Global System for Mobile Communications (GSM), the median positional error is reported to be 5–8.5 m [10] and operates at only 1–2 Hz. Orientation errors from magnetometers are within a few degrees [11], but are heavily affected by external factors such as the proximity of ferromagnetic materials [12]. This performance is clearly insufficient for the demands of fine-grained annotation. Feiner *et al.* [1] already stated in 1997 that, due to inaccurate sensors, it is only possible to assign annotations to buildings and not to fine details (e.g., windows, advertisements).

In contrast, computer vision algorithms are mature enough for localization and tracking with high accuracy and performance. Unlike nonvisual sensors, camera hardware and computational capabilities of mobile devices are rapidly improving. In the following, we give an overview of new computer-vision-based techniques for outdoor registration in next-generation AR browsers. We organize the section based on whether prior knowledge on the environment is available, and we also consider the combination of computer vision techniques with nonvisual sensors.

B. Registration Without Prior Knowledge

Recent results in simultaneous localization and mapping (SLAM) show that it is possible to determine the camera pose in 6DOF without any prior scene knowledge (e.g., fiducial markers placed in the environment or 3-D laser scans of the scene). Because such prior knowledge is often not available, 6DOF SLAM can greatly extend the range of situations where AR can be used. However, 6DOF SLAM has disadvantages: besides high computational requirements, it introduces the need to walk long distances to obtain reliable measurements of large physical structures such as buildings and is prone to error accumulation. While 6DOF SLAM is possible on mobile devices [13], it is restricted to small-scale environments and thus mostly used indoors.

To make outdoor registration feasible, one can reframe the problem of precise tracking in unknown scenes by combining high-resolution orientation tracking with low-resolution position tracking (through GPS). This is justified by the fact that, in outdoor environments, the rotational error contributes more to the perceived error than the positional offset [15].

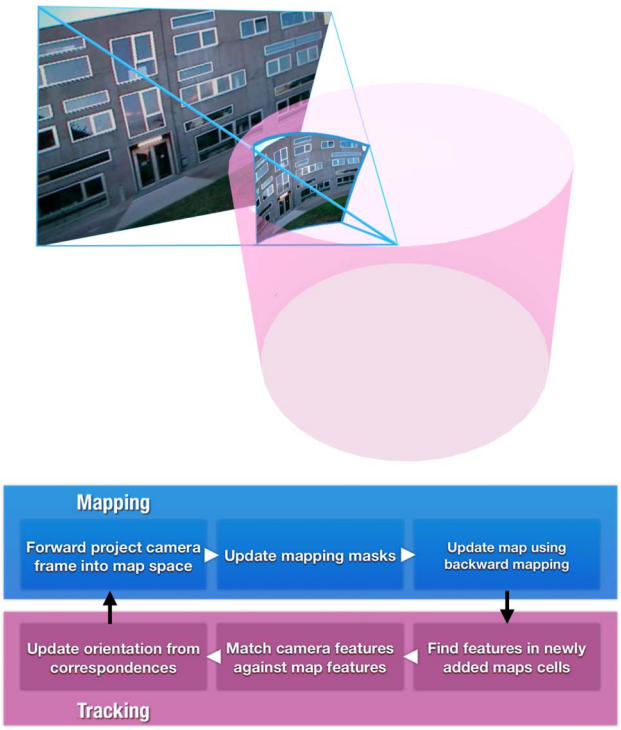


Fig. 4. *Overview of the panoramic mapping and tracking approach. (Top) Projection of the camera image into an cylindrical mapped panorama. (Bottom) Overview of the steps involved in mapping and tracking using a panoramic image [14].*

Based on this idea, we developed a computationally lightweight technique for panoramic tracking and mapping [14] restricted to only 3DOF (Fig. 4). Rather than using a traditional 3-D space, we model the environment as a panoramic space. This approach assumes only rotational motion while standing in the same location, a behavior which is typical for an AR user seeking information [2]. Small translational movements can be neglected. While the panoramic image map is created, we automatically extract feature points and use them to track the relative orientation of the next camera frame. Panoramic mapping and tracking can thus deliver high-quality orientation tracking for mobile AR browsers.

This mapping and tracking runs in parallel and can consequently be seen as a 3DOF SLAM approach. Building the map of the environment has the advantage of not being prone to drift due to an accumulating error such as when using optical flow for determining rotations [16]. In an early approach [14], we keep adding new regions directly onto the map while also applying masks to identify areas that are already mapped; see details in Fig. 4. In the later approach, similar to the work presented by Kim *et al.* [17], we update the panoramic map through adding selective key frames to cover the map of the environment. While the

former approach has advantage in low memory footprint and computation efficiency, the latter approach opens possibilities for a generalized 6DOF tracker and the usage of bundle adjustment to improve accuracy.

SLAM approaches supporting arbitrary camera movement (6DOF) were already presented on mobile phones by Klein and Murray [13] and achieve real-time performance as well. However, they have been so far only truly usable in small spaces (a table or a corner of a street) as managing large spaces (e.g., a city, a building) needs more advanced techniques to manage the construction and to update the general maps.

C. Registration With Prior Knowledge

Having prior knowledge of the real world provides significant advantages. Registration is generally more robust, suffers much less from error accumulation compared to SLAM, and pose measurements are available in real-world units rather than using arbitrary scales. Most importantly, virtual content can be placed in a commonly agreed global coordinate system.

In small-scale environments, it may be feasible to place well-known physical objects such as fiducial markers, or to provide the system with a database of easily recognized natural features such as on the pages of a magazine or a board game [18]. Obviously, outdoor registration requires larger 3-D maps and, consequently, more preparation work.

City scanning projects such as Google Street View can provide large image collections from which a 3-D map of a city can be reconstructed, usually given as a feature database. Registration with 6DOF can be computed by extracting features from a live camera image and matching them against the database.

While this approach, in principle, provides good registration results, it does not scale easily to large databases. On the one hand, downloading and storing large databases on handheld devices is not feasible. On the other hand, a single camera image with a narrow field of view, as is typical for mobile cameras, often does not provide a sufficient number of discriminative features to successfully register the image against a large database containing many, very similar, features.

As a remedy, we have developed an approach that combines panoramic tracking and mapping on a mobile client with registration against a large database on a server [19]. The mobile client builds a panoramic image and uses it for relative orientation tracking. The panoramic image is uploaded to a server for matching, and the registration result is sent back to the client. This overcomes the aforementioned limitations: a panoramic-stitched image contains more features than a single camera image and thus has a higher chance of successful registration. As the user sweeps the camera, more information is incorporated into the panorama, until finally a successful registration is obtained. Moreover, the orientation tracking relative to the panorama overcomes the latency of server–client communication. This approach yields high precision registration while having the advantage of performing real-time localization on mobile devices.

D. Hybrid Tracking

Even with prior knowledge, camera tracking is problematic under rapid motion due to motion blur. Fortunately, a compass, an accelerometer, and a gyroscope available in mobile devices are complementary to the camera—they lack accuracy but are robust against fast motion. We can, therefore, use sensor fusion to stabilize the orientation tracking. When the computer vision algorithms suffer from reduced accuracy due to blur, more trust is placed in the other sensors, so that the overall system performance is stabilized. This is successfully demonstrated in our work [11], [20] on mobile platforms when combined with the presented panorama-based trackers.

We employed a multiple sensors fusion approach which uses vision-based camera tracking together with commonly available sensors, including a differential GPS, a compass, a magnetic, an accelerometer, and a gyroscope. Through strategically integrating multiple sensors and vision-based tracking, we were able to achieve highly robust tracking systems with high accuracy with both going beyond the systems relying solely on one approach.

E. Future Work

Even though our current approach breaks the boundaries of small-scale AR setups, it still incurs a tremendous preparation effort. Gathering millions of images covering an entire city requires massive human and computational resources, yet the resulting database is static and does not reflect the changing nature of the urban environment.

While current city-scale reconstruction projects are led by large companies, we observe a trend toward user-created databases. OpenStreetMap provides high-quality 2-D maps through crowd sourcing, and collaborative 3-D mapping prototypes are beginning to emerge [21], [22]. In the future, users may enhance databases as a side effect of supplying images for registration to the servers.

Beside integrating user-generated images, we can also leverage legacy geodatabases such as cadaster or tagged photo collections. For instance, recent work by Arth *et al.* [23] demonstrates registration directly to geolocated image collections. Robustly incorporating various sources of prior information can be difficult, but it will be essential for providing scalable AR registration.

Nonetheless, output from registration should create consistent and meaningful information for users (position of objects, type of objects, etc.). Therefore, further necessary steps should include the enhancement of the meaning of geometric 3-D maps. Recent trend in 3-D reconstruction shows the possibility to make the 3-D maps at the object's level [24]–[26]. Although these works are limited to selective objects and tabletop or room size environments, the potentials and implications of these methods create

interesting future work. It opens up new possibilities to extend our current registration techniques in order to produce meaningful and semantic output.

IV. CONTENT

To the end user, AR is a new medium, and, thus, it relies on content. Rich content should incorporate not only text, but also pictures, 3-D models, audio, and video. Making content accessible on AR browsers comprises three main challenges: content creation, content description, and content integration.

A. Background and Challenges

Commercial AR browsers today mostly rely on GPS tagged textual descriptions or pictures. AR browser companies try to act as gatekeepers to the content by using proprietary formats. While web authoring can rely on open standards, AR content developers must target every AR browser separately. While the authoring tools provided by AR browser vendors incorporate certain web standards such as HTML, the content generated by these tools is saved in a custom format in the database of the browser vendor.

The dominant workflow for authoring involves first importing existing content, such as pictures or webpages, and then georeferencing it via a web-based map interface. Today, some AR browsers, such as Layar, also promote AR authoring tools for associating printed image targets, such as magazine pages, with virtual content.[2]

Yet, these authoring tools run on desktop computers rather than directly in the mobile AR browser and, consequently, they do not emphasize spontaneous content creation by the crowd. Recently, we have seen mobile authoring tools, such as Aurasma,[3] but they rely on simple object recognition rather than supporting spatial authoring. Obviously, authoring using the map of the world while in one's office cannot support precise registration and lacks the ability to incorporate minute characteristics of the physical environment. Moreover, the supported types of content are rather limited, for example, only predefined layouts and fonts are supported, and importing 3-D models, audio, and video is cumbersome.

More dramatically, users can hardly generate multimedia content for AR experiences directly *in situ* and, without external content editing tools limiting the general usage of social media and constrain it so simple media form. Consequently, the majority of user-generated content for AR browsers consists of simple text and images, while audio and video are hardly used.

One of our research goals has, therefore, been the *in situ* creation of precisely registered rich content from within an AR browser. While we have designed solutions for text, audio, video, and 3-D content, we also had in mind that the interface and the workflow should support social media in AR.

[2]www.layar.com/creator.
[3]www.aurasma.com.

B. Textual Annotations

Textual annotations are the most common content used within AR browser applications. Despite its simplicity, there are no existing solutions that allow creating and placing annotations from within an AR browser. Text associated with a POI is only coarsely registered because the location is given at the accuracy of GPS coordinates. Better registration accuracy for POI annotations can be achieved by incorporating computer vision for recognizing the POI [27].

This goal can be achieved in a very economic way suitable for mobile devices by building a panoramic mapping and tracking in combination with GPS. Users produce and consume annotations on POIs visible from a particular location, for which GPS coordinates are recorded. While this matches the behavior of users selecting "scenic" locations, detection of a POI only works when the user is close to the position from which the annotations were created. Therefore, we added a 2-D map interface (see Fig. 5, left) to guide a user to these spots. Once in the proximity, the user can switch to the AR interface (see Fig. 5, right).

Detection of the POI relies on image patches recorded around the image location designated by the user when creating an annotation. The annotated image patches are stored on the server along with the annotation and the user's current GPS coordinate. When another user later wishes to display nearby annotations, the image patches are retrieved from the server, after filtering by the current GPS coordinate. Rather than searching for image patches in the live video stream, we run the panorama mapping and tracking in the background, and compare patches to the panorama as it is created.

To better cope with varying illumination conditions, we perform matching on images with extended dynamic range. We further constrain the matching using the built-in orientation sensor. Finally, a global rigid rotation is estimated that minimizes the overall error between recorded and current positions of annotations [11]. With these measures, detection rates are almost 90% in realistic outdoor conditions, despite varying environmental conditions and limited computational power on mobile devices. Furthermore, the precise placement contributes to a more seamless integration of the content into the environment. However, this approach has still the downside that this experience is only available for static positions, which require guidance toward them.

C. Audio Annotations

Whereas many AR applications aim for a visual overlay of digital information, audio AR is rarely used. The concept of audio AR was introduced by Bederson in 1995 [28], bringing the idea of adding aural information integrated into our physical environment. Similarly to textual

Fig. 5. *Overview of the prototype demonstrating precisely anchored textual annotations in AR. (Left) Two-dimensional map overview showing spots with attached annotations. (Right) AR interface showing nearby textual annotations that are precisely matched against a panorama built in a background process.*

annotations, audio information is tagged and attached to the GPS position when used in outdoor environments. An example is the work presented by Rozier *et al.* when demonstrating an AR browser's prototype for audio annotations [29]. Audio AR does not suffer from imprecise registration as much as visual AR, since human auditory resolution is lower than visual resolution. However, audio content is mainly experienced in the temporal domain, and thus multiple audio sources playing at the same time are difficult to understand [30], leading to audio clutter.

To address this problem, we developed a new type of audio annotations which are precisely registered in the spatial domain: audio stickies are placed similarly as textual annotations, and other users can play individual audio stickies by pointing at them. The combination of visual selection with auditory display allows a higher density of audio annotations.

Again, we rely on panoramic mapping and tracking for determining the placement of audio stickies. The user simply selects a suitable location on the touch screen and records an audio comment. The recorded comment is saved and referenced to the specified location using the same approach as for textual annotations.

Users can share their own audio stickies and listen to the audio annotations created by others. Audio stickies available at the current location are indicated using visual icons and can be selected via a crosshair. We play only those sounds for which the distance to the focus point is below a certain threshold (see Fig. 6) and adjust the individual volumes so that, at most, two sound sources are played loudly.

The distance to the focus point is also used for creating a spatial audio effect with the stereo panning technique. During audio play, the visual hints, represented by colored dots, used a visual feedback for current states of the audio annotations. For instance, the dots turn green if the audio is about to be selected to play or currently playing. On the other hand, red dots are inactive audio annotations.

D. Video Annotations

The availability of inexpensive mobile video recorders and the integration of high-quality video recording capabilities into smartphones have tremendously increased the amount of videos being created and shared online. With more than 50 hours of video uploaded every minute on YouTube and billions of videos viewed each day, new ways

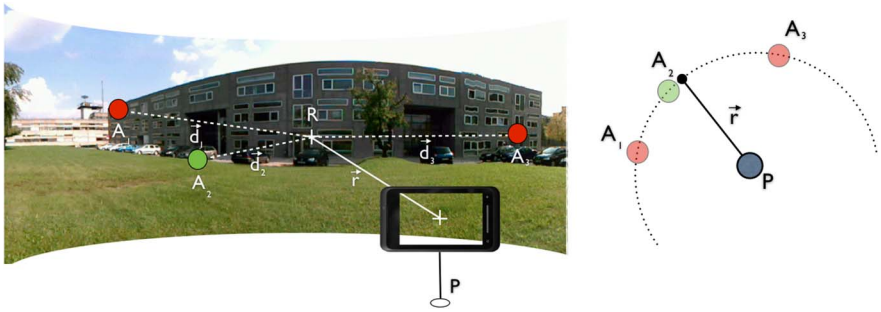


Fig. 6. *Illustration of the panorama-mapped audio stickies. (Left) User at position P browses audio annotations (A_1, A_2, A_3) in the environment. The current focus point R is determined by casting a ray r from screen center onto the panorama of the environment. The volume and the position in the stereo channel of each audio annotation are determined by analyzing vectors (d_1, d_2, d_3) pointing from the focus point to each audio annotation. (Right) Top down view illustration.*

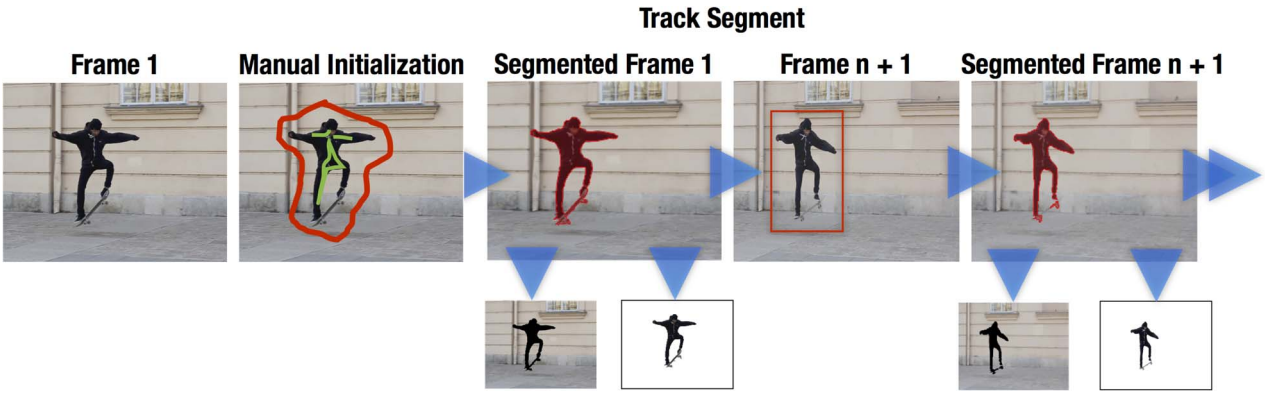


Fig. 7. *Overview of the user-initialized video segmentation. The user sketches the foreground object (green) and outlines the background (red) for initializing GrabCut-based segmentation. Subsequent video frames are segmented by tracking the segments using Lukas–Kanade tracker and using the predicted segments for initializing GrabCut.*

to search, browse, and experience video content are highly relevant.

Video content in AR browsers could be attached to any flat surface, such as pages of a magazine or a building facade.

Most AR browsers simply let video content float freely in space. A more seamless integration turns flat surfaces in the environment, such as pages of a magazine or a building facade, into a video display. However, truly seamless integration of video into AR would use video compositing: a live action foreground (such as a performing actor) is extracted from a static background and inserted into a different environment. Video compositing is commonly used in feature movie production, but not yet in AR.

MacIntyre *et al.* [31], [32] showed how videos can be acquired offline using conventional chroma keying and then composited into real-time AR experiences. However, we were interested in going one step beyond that by also capturing the videos directly *in situ* on mobile devices with only minimal user input and preparation.

Our system, named AR Record & Replay, operates in three steps [33]. In the first step, the video is captured, while the user is allowed to make rotational movements. In the background, we run the panoramic tracking and mapping and also record the GPS location.

In the second step, we segment the foreground object in the video frames using GrabCut [34]. For the segmentation initialization, the user has to roughly sketch the foreground object and mark some background pixels (see Fig. 7). The following frames are segmented automatically by propagating the segmentation of the current frame using optical flow, with the result refined by GrabCut [35] (see Fig. 7). Local adjustment can also be done manually for specific frames to improve the quality.

The background image, with a removed foreground (e.g., black silhouette of skater), is used as an input to build a panoramic map. In practice, we deactivate the pixels corresponding to the foreground object in the background image to build the panorama, creating overtime a seamless background (e.g., uniform wall).

In the third step, the user can replay AR video at the same location, but with free control of the camera viewing direction (Fig. 8, left). This requires running the panorama mapping and tracking for relative orientation tracking and establishing the transform from the background panorama created in step two with the current panorama. In practice, we match them together using a point feature technique (Phony SIFT [18]), which provides us the transformation describing the relative motion between the camera used to record the video (the source camera) and the mobile camera used for the AR view.

This approach allows us to rotate the target camera completely independently from the orientation of the source camera and to maintain the precise registration of the video in the current view (see Fig. 8, left).

Because of the nature of our approach, we are able to perform a wide variety of video effects in real time on a mobile device without the need to pre-render the video, such as overlaying multiple videos or space-time effects (Fig. 8, right).

E. Three-Dimensional Media

The adoption of 3-D media in AR seems to be mostly hindered by complex 3-D modeling tools, which are only available on the desktop. Within our research on *in situ* creation and usage of 3-D media in AR browsers, we identified two core challenges: first, enabling nonexpert users to create 3-D media and, second, allowing the creation process to be *in situ*.

We especially aimed at scenarios allowing to easily duplicate existing geometry in the physical environment of the user such as buildings or other physical objects.

Therefore, we aimed to let a user create 3-D objects using simple touchscreen gestures. To generate 3-D primitives, we rely on a simple interaction mechanism using a

Fig. 8. *Output when replaying videos using our approach on an iPhone4. (Left) Playing back two video augmentations via switchable layers. (Right) Flash-trail effects applied in real time visualize the path and the motion within the video.*

step-by-step process and visual feedback [36]. Using the coordinate system determined by sensor-assisted panorama mapping and tracking, an intersection with the ground can be determined by raycasting. Thus, the user can trace the footprint of a polygonal model and extrude it to form a 3-D model (Fig. 9). The supported 3-D models range from cubes, tubes, and spheres to objects with arbitrary polygonal ground planes [36].

Objects can be colored and textured. The user, therefore, can select from a set of predefined textures, which can be mapped to the object or create new textures by selecting a region of the current camera image, which can later be used as texture and assigned to objects (Fig. 9, right). This allows rapid capturing of real objects, such as buildings, using simple bounding geometry.

F. Future Work

Overall, we showed several approaches that demonstrate a seamless integration of rich media AR browser applications through precise registration within the environment. While current AR browsers use mostly textual annotations, we emphasize our call for rich media in next-generation AR browsers by presenting solutions for various media, including textual, audio, video, and 3-D information. Our experimental AR browser supports not only rich media, but also casual users in creating all these types of content directly in the AR browser and *in situ*. This approach to social AR lowers the threshold for creating content for AR and hopefully contributes to a more dense distribution of situated media accessible in AR.

Future research is still required in interfaces and designs, further easing the creation process, especially when creating high-quality content. Other issues that need to be solved are the demand for open formats for AR browsers that can be used to describe content and its position. While current-generation AR browsers rely on different formats using one position system (GPS), we need one format incorporating different position systems (e.g., sensor- as well as vision-based approaches).

V. USER INTERFACE

AR interfaces can be seen as interfaces that allow to interface and interact with (2-D or 3-D) virtual content placed on a physical 3-D space or simple physical objects. Interaction in AR browsers can range from browsing textual annotations overlaid on a video view of the physical world to playing a range of simple games (e.g., scavenger, puzzle) in an outdoor environment or playing interactive media on augmented objects. Yet, the current user interfaces proposed in the AR browser are limited, and generally only provide a non-seamless visual presentation of the content within our physical world with a limited range of interaction techniques.

Fig. 9. *Example showing augmented 3-D content using the system in outdoor environments. (Left) Real scene as displayed in the camera view of a smartphone, (Middle) Augmented scene showing the created virtual object (highlighted in green) during placement operation. (Right) Final scene showing a created duplicate of an existing building augmented to the left of the real building.*

A. Background and Challenges

As MacIntyre *et al.* [37] pointed out, we see a shortcoming in interaction techniques in AR browsers and other AR applications. Apart from a viewpoint manipulation, that interaction is mostly limited to clicking on virtual hotspots and is not concerned with physical content. Engaging AR experiences should allow manipulation of physical content and its relation to virtual content. This requires new conceptual models in AR interaction, which consider both physical and virtual content in the same framework.

AR browsers should enable interaction in two ways: they should consider a larger variety of objects in our physical world, and they should provide a unified interface. For the former, the environment, places, or people should be recognized and integrated into this type of system, and react to their properties (e.g., surface of a building). For example, Takeuchi and Perlin [38] demonstrated how AR can be use to mediate between physical objects, such as rescaling them or applying nonlinear transformation (e.g., bending). This type of approach can be envisaged to be integrated into an AR browser. For the latter, we can draw from a rich legacy of interaction techniques in conventional applications, both on the desktop and on mobile devices. Games, modeling tools, and even office applications offer much richer interaction than current AR browsers.

B. Rethinking the Layout: View Management

Traditional desktop interface ["windows, icons, menus, pointer" (WIMP) interfaces] or Web user interface (implemented in HTML) provides us guidance on how content should adapt and considers placement of elements in relation to each other in an AR view. Existing standardized layout techniques for HTML DOM or for adjusting window management in the WIMP system should be applied in the context of the AR browser.

To address this aspect, Bell *et al.* introduced the concept of *view management* for AR [39]. The main idea was to go beyond the traditional AR registration pipeline by managing virtual content as the function of measured or observed characteristics of our environment: dynamic physical objects, knowledge of 3-D building, to adapt the *layout* of the content (in 2-D or 3-D) as well as its *representation* (chrominance, luminance, etc.). In their work, they demonstrated a proof of concept for textual information while relying on *a priori* knowledge of the environment (a 3-D geometric scene).

Similarly, Kooper and MacIntyre proposed, within their real World Wide Web [4], to adapt the representation of labels based on the context of the user. Using gaze selection, virtual documents overlaid on an AR view change their representation from an iconographic mode to a thumbnail mode or a full representation of a document.

A more generalized approach was presented by Julier *et al.* under the notion of *adaptive user interface* [40], capturing contextual aspects of a scene, filtering measured information, and providing AR user interfaces which adapt to the context. AR browser technology has currently only a limited implementation of this concept: filtering the content based on the distance between POI and the user's location. Yet, view management techniques as well as adapting any type of a user interface (e.g., size of annotations) are absent.

Designing and developing adaptive user interface is certainly one of the major challenges in AR for the next few years. Over the last years, we have been investigating the arguably most important type of contextual information for adaptive user interfaces: content of the real-world view, i.e., the video image. We started with exploring new layout and representation techniques for the most popular types of POI, textual labels.

1) Image Importance: Our first approach was focused on placing labels so that important real content in the video image was preserved [8]. For example, a label should avoid dynamic elements (e.g., car moving toward you) or a salient region (e.g., physical signs, cultural artifacts, etc.). Instead, label placement algorithms should focus on uniform areas such as the sky or a uniform façade in the view on the physical world (e.g., video image) as well as highly repetitive and fine grain structure such as grass that does not yield important information to the user.

We approached this problem by associating the notion of importance in a view to salient content of an image (Fig. 10). We used the existing saliency computational model [41] and based our approach on a saliency map: dark areas on the map correspond to the nonsalient area, while bright areas correspond to highly salient elements.

The main idea of the work was to use this saliency map as an input to a layout optimizer, which adjusts the position of labels based on this input (i.e., it moves labels to nonsalient regions). A greedy algorithm integrates standard layout factors such as avoiding label overlap, label leader line crossing, etc. Our technique also made use of an edge map as an additional input to give appropriate weight to objects with high spatial frequencies, such as zebra crossings or power lines.

We tested the system in a range of urban and rural scenarios (Fig. 11). An additional challenge in AR is to handle temporal continuity and the moving camera: labels should not continuously change position as a result of small camera movements. Based on our recent analysis of user behavior within an AR browser [2] and our observation from a decade of user studies in AR, we empirically determined three relevant types of motion: navigation motion (e.g., rotating the AR device), image motion (e.g., car moving in the image), and jitter (nonvoluntary motion due to hand jitter). Depending on the motion type, we trigger an adaptation of the position of the label (in case of navigation motion) or decide to keep the current label position (in case of jitter). Using this approach, we want

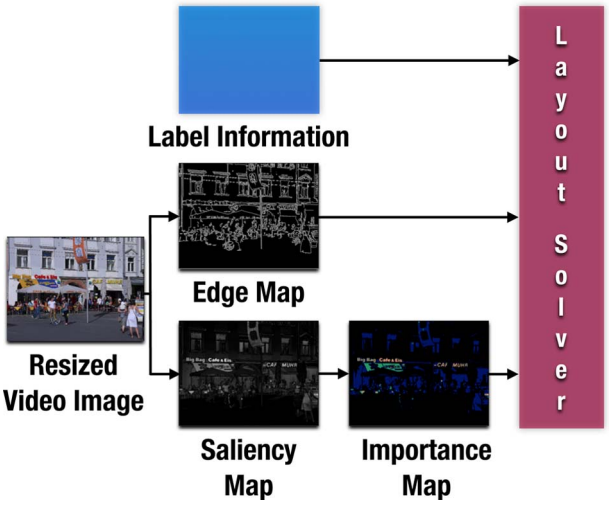


Fig. 10. *Algorithm using the importance of image areas for the optimization of label layout: from an initial video image (left), we compute an edge and saliency map (middle) and combine it with label information (initial position, representation policy), using this information in a layout optimizer (right), resulting in updated location and representation parameters for displaying the labels.*

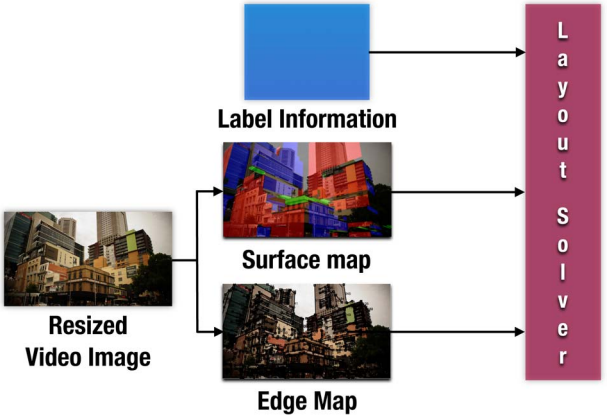


Fig. 12. *Algorithm using the geometry within the image for the optimization of the label layout: from an initial video image (left), we compute a surface map and an edge map (middle) and combine them with label information (initial position, representation policy) using this information in a layout optimizer (right). The optimizer provides new orientation and representation for displaying the labels.*

to reduce the effect of constant and unnecessary label position updates that negatively affect the overall experience.

2) Image Geometric Structure: In the following work, we investigate using the *geometric structure* in a video image for view management. Our main idea is to recover relevant spatial information from an image to inform label placement and orientation (Fig. 12). Starting from initial edge detection, we compute the vanishing points in the image and use this information to compute the orientation of vanishing planes, a *surface map* (or a *vanishing map*). As shown in the left image of Fig. 13, each color of generated geometric primitives corresponds to a vanishing direction. The projected position of the POI is used to detect the vanishing plane to which the label should be geometrically aligned. If the label does not intersect any vanishing planes, the closest region or the closest edge (e.g., top of a building) is used (Fig. 13).

We are currently investigating additional techniques for using the vanishing map. For example, a layout policy can demand that a label be parallel or perpendicular to the vanishing planes, always snap to edges, etc. It also seems possible to infer approximate volumetric information on the scene from vanishing information and use it for view management.

C. Unifying the Interface: Web Technology

One way to offer a more complete, flexible, and extensible access to interactions with AR content is via web technology. The success of Web 2.0 can be explained by its simplicity and extensibility (i.e., Javascript), which resulted in a large and active community of developers. How can we reproduce such a platform for AR browsers?

Fig. 11. *Example of applying image importance to the view management. (Left) Original image with labels placed based on their position. (Right) Results of our optimized label placement based on the importance map determined in real time from the camera feed. The small insets show the importance map and the edge map.*

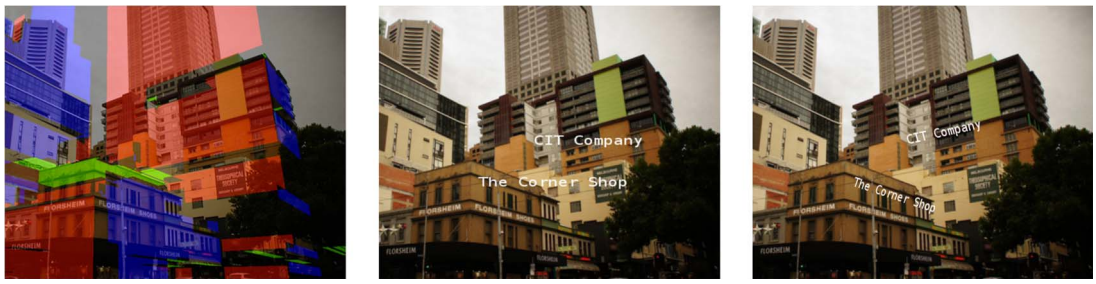


Fig. 13. *Example of applying geometric constrains determined in real time from a single image. (Left) The computed surfaces and their orientation within the image. (Middle) The labels placed at their original position. (Right) Applying a layout algorithm based on geometric constrains such as the surface map and the edge map allows to automatically align the labels with their underlying geometry depicted in the current camera image.*

Recent AR browsers have slowly taken up web standards, but in a really limited way. Recent formats such as ARML, KARML [7], or Layar's JSON format[4] do not yet provide a standardized platform. Similarly, 3-D rendering uses proprietary 3-D rendering engines rather than WebGL or X3D.[5] However, we argue that unified user interfaces for AR would benefit from providing access via DOM and Javascript for unified access to low-level AR input (tracking, sensors) and output (screen, head-mounted display) as well as the view management and other computationally intensive tasks.

In a recent effort toward this goal, we investigated how we can define a generic interface to a low-level AR input by extending the DOM *navigator* object. Multiple tracking components can be queried from the navigator object based on built-in trackers in a web browser or via plug-ins provided by third parties. These trackers could cover planar objects, SLAM trackers, or face trackers.

We implemented a prototype for demonstrating an HTML-based AR interface using the PhoneGap middleware toolkit[6] and the Vuforia Tracker[7] on the iOS platform. We extended the PhoneGap plugin with an AR plugin (ARGap) to provide access to the pose information from the tracker through Javascript. So far we only integrated the Vuforia tracker, but others, including multiple trackers, can be supported simultaneously. The appropriate tracker is selected by querying it via an *ad hoc* DOM navigator object.

Combined with WebGL, this approach allows users to specify 3-D content and interact with it (Fig. 14). Tracking events, such as position updates or out-of-range information, are exposed via a standard interface and can be queried in user interfaces.

For the view management, we employ cascading style sheets (CSSs) to support the definition of the layout and the representation of AR content. AR input devices (e.g., camera, sensors) can be matched with specific DOM objects such as textual annotations or 3-D objects and specific layout or representation policies. For example, a user can specify the luminance of an AR label based on a reading from the ambient light sensor:

```
<div AR2DLabel-luminance="ambient">
    <p>example of a label</p>
</div>
```

Using other modes such as camera-based luminance adjustment or constant values, developers will able to automatically specify an adaptive technique for AR content via simple CSS.

Using jQuery,[8] we were able to parse and identify additional custom CSS rules, and process them using deriving classes for different types of objects and different properties. In future implementations, rules can be directly integrated in an AR browser, and CSS should provide an extension mode to add complimentary policies to an existing rule (e.g., label layout dependent on content of the view) or the possibility to define one's own algorithm for specific properties (e.g., controlling color and luminance of an object simultaneously).

D. Future Work

Better scene understanding will improve seamless integration of virtual content into AR browsers. For example, existing 3-D world data sets (e.g., Open Street Map, Google Earth) would benefit from additional online image analysis in order to improve the presentation of virtual content. Built-in sensors from the device (e.g., measuring a user's velocity, detecting the presence of RFID-tagged objects) can further improve adaptive user interfaces.

Content representation also needs to become more adaptive. For example, readability of labels should be ensured, as proposed in [8], [42], and [43]. Visual coherence [44] could also be addressed in a CSS configurable way.

[4]www.layar.com.
[5]www.web3d.org/x3d/.
[6]www.phonegap.com.
[7]www.vuforia.com.
[8]www.jquery.com.

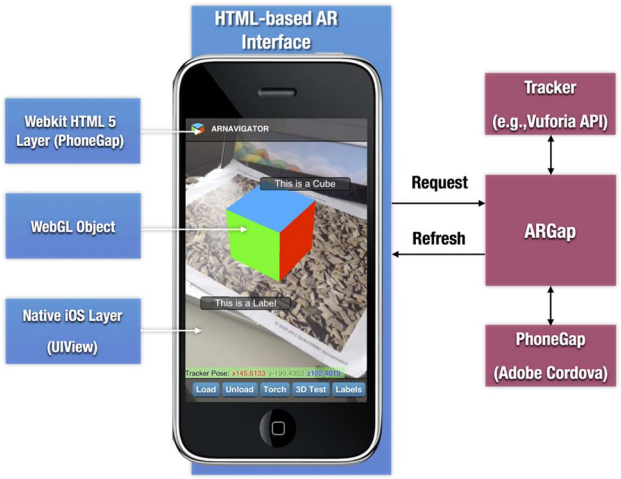


Fig. 14. ***Overview of the architecture of our prototype implementing an AR interface build on HTML.***

Table 1 AR Browser Research Roadmap

Topic	Futuristic Outlook	Research Roadmap
Audience	Global, mass adoption	Social studies, application-oriented prototyping tools and methodologies
Device	Hybrid: handheld, HMD, wearable	Multi-devices user interfaces, distributed AR computing, Quality of Service
Registration	sSeamless, adaptive, hybrid	Large scale cloud-based infrastructure, ubiquitous tracking,c ontextual scene understanding
Content	Rich, seamless, participative, social	In-situ content authoring, multi or cross-modal content, social user interface model, HTML integration
User Interface	Seamless, adaptive, contextual	Spatial view management techniques, in-situ real-time scene analysis for contextual user interfaces, machine learning techniques, HTML-based AR frameworks and tools

Rich content is likely to increase clutter and requires better filtering. For example, content can be clustered or semantically aggregated. It seems likely that visualization techniques currently investigated for "big data" can be useful for this purpose.

Another remaining challenge will be to provide a user interface for unknown situations. While, for example, ambient luminosity can be always measured, the presence of an object appearing in the real world (a car or a person) may not be foreseen at the time of implementation. Reasonable adaptive behavior in such cases is still a great challenge.

VI. DISCUSSION

Pitfalls of current AR browsers, including disconnected tracking technology, static content, and limited user interfaces, can be mitigated with the techniques and future research directions elaborated on in this paper. Supporting rich, seamless, and adaptive content can give rise to the next generation of AR browser.

Developers of AR browsers need to move away from their principal interest in replicating application structures of existing information systems. Instead, they should consider more systematically how the physical world can become a part of the AR user experience. By considering physical objects, places, and people in the real world, they will be able to motivate users to employ AR browsers in real life.

The reader should note that current-generation AR browsers have been designed for handheld devices. Upcoming head-mounted displays will certainly redefine some of the characteristics of AR user interfaces. Some of the work presented here will still apply in this new form factor, but new solutions will evolve. For example, touchscreen interaction will need to be rethought on a hands-free platform. Selecting and navigating will require new concepts, such as eye gaze [4].

AR browsers also need to embrace social computing. Investigating how we can enrich our physical world with knowledge from social networks such as Facebook, Twitter, or FourSquare remains to be investigated. Sharing content and collaborating remotely needs new tools for user awareness in relation to the physical world.

Finally, AR browsers are still a small market with a limited user group, especially when compared to mobile games or social networks. Our recent survey [2] outlined some of the issues addressed in this paper as potential reasons for slow adoption. Important usability aspects of AR technology, such as the device display form factors (handheld versus HMD) still need to be studied. Field studies remain difficult until we have a more stable AR technology and better evaluation tools.

To draw an outlook on future research, and to summarize some of the concepts presented previously, we present in Table 1 current research topics related to AR browsers that work toward a vision of being rich, seamless, and adaptive.

VII. CONCLUSION

We have argued that next-generation AR browsers need seamless registration, rich content, and adaptive user interface in order to succeed. We discussed how these aspects affect key AR components, namely registration, content, and user interface.

Seamless registration remains a challenging topic, which requires further research, especially as the other components depend strongly on the quality of the registration in order to

achieve desirable results. In this paper, we presented new tracking technologies to attain this goal, such as panoramic tracking and mapping approach combined with server-based image queries or multiple sensor fusion.

Rich content will play a significant role in the success of next-generation AR browsers. Therefore, a seamless integration of a wide variety of content forms such as text, video, audio, and 3-D content is an essential development. For this matter, we demonstrated examples of precisely placed textual annotations, audio annotations combined with visual modalities, *in situ* authoring, and seamless integration of video content via an AR panoramic tracking as well as authoring and usage of 3-D content.

Rich content will also lead to more challenges in content presentation. Therefore, improved view management and a unified user interface are important elements of next-generation AR browsers.

Finally, we discussed important challenges that must be addressed in order to achieve seamless, rich, and adaptive AR browsers. In our recent work, we focused on image-driven techniques relying on scene analysis to improve the layout and representation of the content for AR browsers. We also introduced how HTML can be currently used to support the definition of adaptive user interface through CSS extension. While more work remains to be done, we believe these challenges can be addressed and resolved within a reasonably near future, hopefully leading to large-scale adoption of AR. ■

Acknowledgment

The authors would like to thank H. Regenbrecht for his input on several of the projects presented in this paper.

REFERENCES

[1] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," *Pers. Technol.*, vol. 1, no. 4, pp. 208–217, Dec. 1997.

[2] J. Grubert, T. Langlotz, and R. Grasset, "Augmented reality browser survey," Graz Univ. Technology, Graz, Austria, Tech. Rep. [Online]. Available: http://www.icg.tugraz.at/publications/augmented-reality-browser-survey

[3] J. C. Spohrer, "Information in places," *IBM Syst. J.*, vol. 38, no. 4, pp. 602–628, 1999.

[4] R. Kooper and B. MacIntyre, "Browsing the real-World Wide Web: Maintaining awareness of virtual information in an AR information space," *Int. J. Human-Computer Interaction*, vol. 16, no. 3, pp. 425–446, 2003.

[5] R. Azuma, "Tracking requirements for augmented reality," *Commun. ACM*, vol. 36, no. 7, pp. 50–51, Jul. 1993.

[6] T. Langlotz, J. Grubert, and R. Grasset, "Augmented reality in the real world: AR browsers—Essential products or only gadgets?" *Commun. ACM*, vol. 56, no. 11, pp. 34–36, 2013.

[7] B. MacIntyre, A. Hill, H. Rouzati, M. Gandy, and B. Davidson, "The argon AR web browser and standards-based AR application environment," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 65–74.

[8] R. Grasset, T. Langlotz, D. Kalkofen, M. Tatzgern, and D. Schmalstieg, "Image-driven view management for augmented reality browsers," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, Nov. 2012, pp. 177–186.

[9] T. Höllerer and S. Feiner, "Mobile augmented reality," *Telegeoinformatics: Location-Based Computing and Services*, H. A. Karimi, Ed. Boca Raton, FL, USA: CRC Press, 2004, pp. 1–39.

[10] P. A. Zandbergen and S. J. Barbeau, "Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones," *J. Navig.*, vol. 64, no. 3, pp. 381–399, Jul. 2011.

[11] T. Langlotz, C. Degendorfer, A. Mulloni, G. Schall, G. Reitmayr, and D. Schmalstieg, "Robust detection and tracking of annotations for outdoor augmented reality browsing," *Comput. Graph.*, vol. 35, no. 4, pp. 831–840, Aug. 2011.

[12] G. Schall, A. Mulloni, and G. Reitmayr, "North-centred orientation tracking on mobile phones," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2010, pp. 267–268.

[13] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proc. 8th IEEE Int. Symp. Mixed Augmented Reality*, 2009, vol. 41, no. 1, pp. 83–86.

[14] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," in *Proc. IEEE Virtual Reality Conf.*, Mar. 2010, pp. 211–218.

[15] S. You, U. Neumann, and R. Azuma, "Orientation tracking for outdoor augmented reality registration," *IEEE Comput. Graph. Appl.*, vol. 19, no. 6, pp. 36–42, 1999.

[16] D. Wagner, T. Langlotz, and D. Schmalstieg, "Robust and unobtrusive marker tracking on mobile phones," in *Proc. 7th IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2008, pp. 121–124.

[17] H. Kim, G. Reitmayr, and W. Woo, "IMAF: In situ indoor modeling and annotation framework on mobile phones," *Pers. Ubiquitous Comput.*, vol. 17, no. 3, pp. 571–582, 2013.

[18] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proc. 7th IEEE/ACM Int. Symp. Mixed Augmented Reality*, Sep. 2008, pp. 125–134.

[19] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg, "Real-time self-localization from panoramic images on mobile devices," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 37–46.

[20] G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, and B. Hofmann-Wellenhof, "Global pose estimation using multi-sensor fusion for outdoor augmented reality," in *Proc. 8th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2009, pp. 153–162.

[21] W. Yii, W. H. Li, and T. Drummond, "Distributed visual processing for augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2012, pp. 41–48.

[22] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "A component-based approach towards mobile distributed and collaborative PTAM," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2012, pp. 329–330.

[23] C. Arth, A. Mulloni, and D. Schmalstieg, "Exploiting sensors on mobile phones to improve wide-area localization," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 2152–2156.

[24] A. Dame, V. A. Prisacariu, C. Y. Ren, and I. Reid, "Dense reconstruction using 3D object shape priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1288–1295.

[25] Y. Bao, M. Chandraker, Y. Lin, and S. Savarese, "Dense object reconstruction with semantic priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1264–1271.

[26] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1352–1359.

[27] T. Langlotz, D. Wagner, A. Mulloni, and D. Schmalstieg, "Online creation of panoramic augmented reality annotations on mobile phones," *IEEE Perv. Comput.*, vol. 11, no. 2, pp. 56–63, Feb. 2012.

[28] B. B. Bederson, "Audio augmented reality," in *Proc. Conf. Companion Human Factors Comput. Syst.*, May 1995, pp. 210–211.

[29] J. Rozier, K. Karahalios, and J. Donath, "HearThere: An augmented reality system of linked audio," in *Proc. Int. Conf. Auditory Display*, 2000, pp. 63–67.

[30] Y. Vazquez-Alvarez, "Designing spatial audio interfaces for mobile devices," in *Proc. 12th Int. Conf. Human Comput. Interaction With Mobile Devices Services*, Sep. 2010, pp. 481–482.

[31] B. Macintyre, M. Lohse, J. D. Bolter, and E. Moreno, "Ghosts in the machine: Integrating 2D video actors into a 3D AR system," in *Proc. 2nd Int. Symp. Mixed Reality*, 2001, pp. 73–80.

[32] B. MacIntyre, M. Lohse, J. D. Bolter, and E. Moreno, "Integrating 2D video actors into 3D augmented reality systems," *Presence Teleoper.*, vol. 11, no. 2, pp. 189–202, 2002.

[33] T. Langlotz, M. Zingerle, R. Grasset, H. Kaufmann, and G. Reitmayr, "AR Record&Replay," in *Proc. 24th Austral.*

Computer-Human Interaction Conf., Nov. 2012, pp. 318–326.

[34] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004.

[35] J. Mooser, S. You, and U. Neumann, "Real-time object tracking for augmented reality combining graph cuts and optical flow," in *Proc. 6th IEEE/ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 145–152.

[36] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg, "Sketching up the world: In situ authoring for mobile augmented reality," *Pers. Ubiquitous Comput.*, vol. 16, no. 6, pp. 623–630, 2012.

[37] B. MacIntyre, H. Rouzati, and M. Lechner, "Walled gardens: Apps and data as barriers to augmenting reality," *IEEE Comput. Graph. Appl.*, vol. 33, no. 3, pp. 77–81, May/Jun. 2013.

[38] Y. Takeuchi and K. Perlin, "Clayvision: The (elastic) image of the city," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 2411–2420. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208404

[39] B. Bell, S. Feiner, and T. Hollerer, "Information at a glance [augmented reality user interfaces]," *IEEE Comput. Graph. Appl.*, vol. 22, no. 4, pp. 6–9, Jul. 2002.

[40] S. Julier, M. A. Livingston, J. Edward, S. Ii, and Y. Baillot, "Adaptive user interfaces for augmented reality," in *Proc. Symp. Mixed Augmented Reality*, 2003, pp. 35–42.

[41] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.

[42] J. L. Gabbard, J. E. S. II, and D. Hix, "The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor augmented reality," *Presence, Teleoper. Virtual Environ.*, vol. 15, no. 1, pp. 16–32, Feb. 2006.

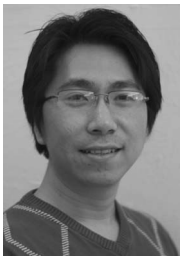
[43] J. L. Gabbard, J. E. S. II, D. Hix, S.-J. Kim, and G. Fitch, "Active text drawing styles for outdoor augmented reality: A user-based study and design implications," in *Proc. IEEE Virtual Reality Tech. Papers*, Mar. 2007, pp. 35–42.

[44] L. Gruber, D. Kalkofen, and D. Schmalstieg, "Color harmonization for augmented reality," in *Proc. 9th IEEE Int. Symp. Mixed Augmented Reality*, 2010, pp. 227–228.

ABOUT THE AUTHORS

Tobias Langlotz (Member, IEEE) received a diploma in media systems from the Bauhaus University, Weimar, Germany, in 2007 and the Ph.D. degree in computer science from Graz University of Technology, Graz, Austria, in 2013.

Since 2014, he has been a Lecturer at the University of Otago, Dunedin, New Zealand. Previously he was a Senior Researcher at Graz University of Technology. His research interests are location- and context-sensitive mobile interfaces at the intersection of human–computer interaction, computer graphics, computer vision, and pervasive computing, in particular, in the field of mobile augmented reality, where he is working on mobile interfaces for situated social media.



Thanh Nguyen (Student Member, IEEE) received the M.S. degree in computer science from the University of South Australia, Adelaide, S.A., Australia, in 2010. Currently, he is working toward the Ph.D. degree in computer science at Graz University of Technology, Graz, Austria.

He is a Research Assistant at Graz University of Technology. His research focuses on real-time computer vision topics, including online interactive modeling and 3-D reconstruction. Derived from computer vision research, he aims to build next-generation registration technologies for augmented reality applications.

Dieter Schmalstieg (Member, IEEE) received the Dipl.-Ing., Dr. techn., and Habilitation degrees from Vienna University of Technology, Vienna, Austria, in 1993, 1997, and 2001, respectively.

He is a Full Professor and Head of the Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria. Since 2008, he has been the Director of the Christian Doppler Laboratory for Handheld Augmented Reality, Graz University of Technology. His current research interests are augmented reality, virtual reality, real-time graphics, 3-D user interfaces, and visualization.

Prof. Schmalstieg is an Associate Editor of the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, member of the Editorial Advisory Board of *Computers & Graphics* and of Springer *Virtual Reality*, member of the steering committee of the IEEE International Symposium on Mixed and Augmented Reality, chair of the EUROGRAPHICS working group on Virtual Environments (1999–2010), and member of the Austrian Academy of Science.

Raphael Grasset (Member, IEEE) received the Ph.D. degree in computer science from Université Joseph Fourier, France, in 2004.

He is a Senior Researcher at the Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria. Previously, he was a Senior Researcher (2007–2010) and a Postdoctoral Researcher (2004–2007) at the HIT Lab NZ, Canterbury University, Christchurch, New Zealand. His main research interests include: 3-D interaction, computer-human interaction, augmented reality, mixed reality, visualization, and computer supported collaborative work (CSCW). His research methodology is both bottom-up (build and evaluating technology and interfaces) and top-down (application driven). He has been involved in a large number of multidisciplinary academic and industrial projects over the last decade. He is the author of more than 50 international publications, and he supervised more than 50 students in the last eight years.

Dr. Grasset has been a member of the International Symposium on Mixed and Augmented Reality (ISMAR) Steering Committee since 2010.