Towards Embedded Visualization Authoring in Augmented Reality

Dieter Schmalstieg
schmalstieg@tugraz.at
Graz University of Technology

Philipp Fleck
philipp.fleck@icg.tugraz.at
Graz University of Technology

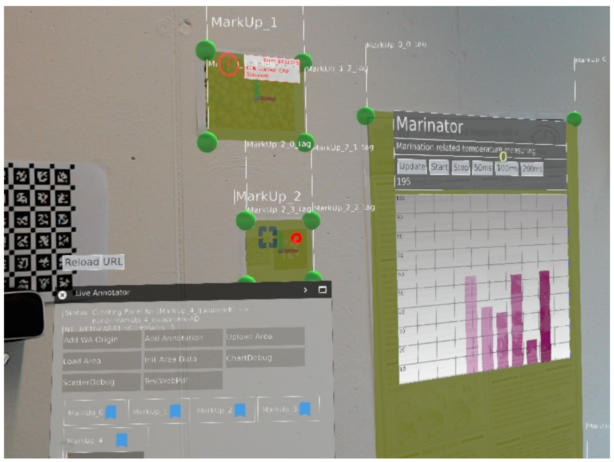

Figure 1: Seen through the Hololens: Canvases (yellow) referenced in the real-world (as overlay to posters on the wall) with attached content like a streaming barchart (having an attached sensor source for input) and two HTML views (top and center).

ABSTRACT

We describe a system designed to empower an analyst to create an embedded visualization in-situ, at a task location, using an augmented reality display. The analyst defines (or lets the system choose) canvases in convenient locations, places data visualization views in these canvases, and connects them to (potentially live streaming) data sources. Our design can deal with the fully dynamic situations that must be considered in maintainance and repair scenarios or other scenarios that must be carried out at a given task location, away from the desktop.

KEYWORDS

immersive analytics, situated visualization

ACM Reference Format:
Dieter Schmalstieg and Philipp Fleck. 2018. Towards Embedded Visualization Authoring in Augmented Reality. In *Immersive Analytics*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

INTRODUCTION

Visualization and visual analytics are rapidly growing fields. However, professional use of these technologies is mostly tied to desktop computing or stationary display ecologies, e.g., as found in

IA 2002, 2020, Honolulu, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

a control room. In contrast, an overwhelming majority of activities involving information systems today is conducted away from the desktop, on mobile devices. Consequently, a huge potential remains unexplored, since conventional workstyles involving visualization and visual analytics are too cumbersome on mobile devices. Currently, this includes Augmented Reality (AR) displays. Users may be willing to *inspect* visualizations in AR [1], but *authoring* of visualizations (or other more complex visual analytics procedures) have been largely elusive so far.

To state the obvious, the expected benefit of combining visualization and AR comes primarily from being able to easily access logical information in a physical context. Analysts can be released of the burden of having to cross-reference physical perception and logical information.

In other words, when using AR, interacting with and modifying individual marks in a visualization (the tasks most commonly associated with visual analytics work) seems less interesting than creating a visualization pipeline, i.e., *embedded visualization authoring*. It is important to note that we mean "embedded authoring of visualizations" and not merely "authoring of embedded visualizations".

In this article, we describe a system designed to empower an analyst to create an embedded visualization using an AR display, such as the Microsoft HoloLens. The analyst visits a task location, for example, a factory floor where machinery requires troubleshooting or tuning. The analyst's objective is to understand the data in its natural context. Today's factories are increasingly characterized as cyber-physical systems (CPS), where an abundance of telemetry data is available from embedded sensors. Accessing such data using either stationary information displays or mobile devices, such as tablet computers, is possible, but not very efficient, at least when a deeper investigation is required.

In the following, we will first review important previous work pursuing this direction and highlight some emergent requirements. These requirements will inform the design of a framework facilitating embedded authoring of complex visualizations in an AR setting.

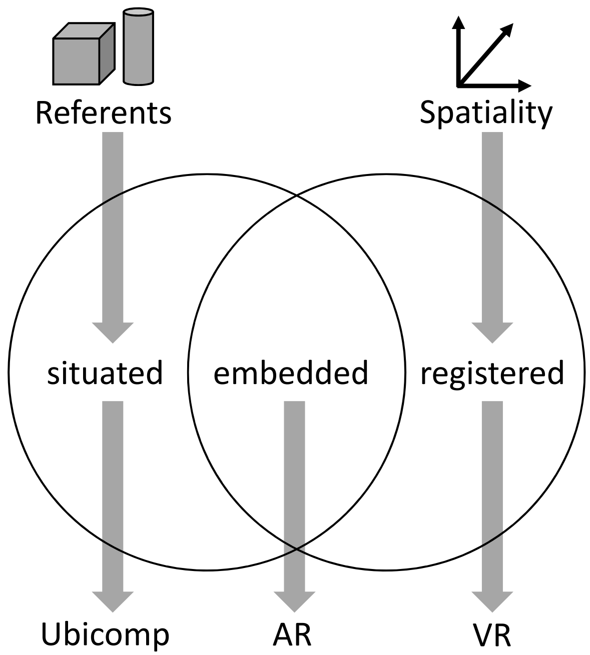

Figure 2: Embedded visualizations are both situated with respect to referents and registered spatially with respect the environment.

RELATED WORK AND MOTIVATION

An essential starting point of our discussion is the terminology introduced by Willet et al. [19] for characterizing *embedded data representations*, which take visualization beyond the desktop. In their notion, "embedded" means (1) *situated* in perceptual proximity to a physical *referent* and (2) *registered* spatially with the user's environment.

This terminology highlights the two essential dimensions of the problem space: *Situated visualization* requires that a user can associate an information display with referents in the environment. This kind of interface can be facilitated by *mobile and ubiquitous computing* (e.g., on smartphones). In contrast, *immersive visualization* requires precise spatial awareness. This need can, for instance, be fulfilled in *virtual reality*, without the involvement of any physical entities. Embeddedness, the combination of situatedness and registration, cannot be addresses by ubiquitous computing or virtual reality alone. It requires *AR*, which is capable of both contextual awareness *and* spatially awareness (Figure 2).

Jansen and Dragicevic [10] discuss architectural foundations for building such embedded data representations. They observe that the traditional infovis reference model is defined within the confines of screen space and ignorant of the physical world. To overcome this limitation, they propose to extend the visualization pipeline with two important concepts: First, they introduce a *percept* stage which models how a user receives the visualization in the physical world, allowing a visualization designer to consider perceptual aspects, such as distance, subtended angle, illumination, contrast and so on. Second, they propose to rely on *instrumental interaction* to enable the visualization designer to incorporate any mixture of physical and logical tools into the interaction. Both percepts and instrumental interaction are powerful ideas, warranting a deeper investigation.

Percepts. A user's percept is created by a physical information display, which Bach et al. [2] call the *canvas*: They define it as the "part of a viewer's field-of-view where information visualization is rendered in-situ with respect to visible and potentially invisible real-world objects". Such a canvas is three-dimensional and often already populated with physical objects, which can be meaningful referents or merely obstacles. Thus, the canvas imposes many perceptual constraints. Moreover, the physical world, which includes canvases and referents, is dynamically changing. Reacting to changes raises the need to continuously update the visualization and, thereby, generalizes traditional AR design problems such as label placement and view management [4].

Instrumental interaction. Beaudouin-Lafon [3] proposes instrumental interaction as generalization of direct manipulation with the intent to make it fit for post-WIMP interaction. His key idea is to introduce a separation of concerns into domain objects and interaction instruments. This separation is somewhat similar to the *model* and *controller* aspects of the model-view-controller (MVC) pattern in software engineering. To best support fluid interaction, domain objects should be given a continuous representation and allow for physical actions, which are fast, incremental, reversible and have an immediately apparent effect.

These physically inspired characteristics of domain objects in instrumental interaction are reminiscent of the *reality-based interaction* (RBI) model of Jacob et al. [9], who argue that interaction design should build on physically grounded cognitive phenomena, such as naive physics, body awareness and environmental awareness. These are properties known from, e.g., tangible user interfaces, but they equally apply to AR. This prompts the question if we can hope to use instrumental interaction for embedded visualization in AR.

Unfortunately, at first glance, there appears to be a mismatch: Complex data abstractions, which are commonplace in visualization, do not always map easily to physical metaphors and can suffer in their representational efficiency when forced to do so. Does this imply the designer has to choose between an efficient abstract representation and a representation that lends itself to RBI or instrumental interaction?

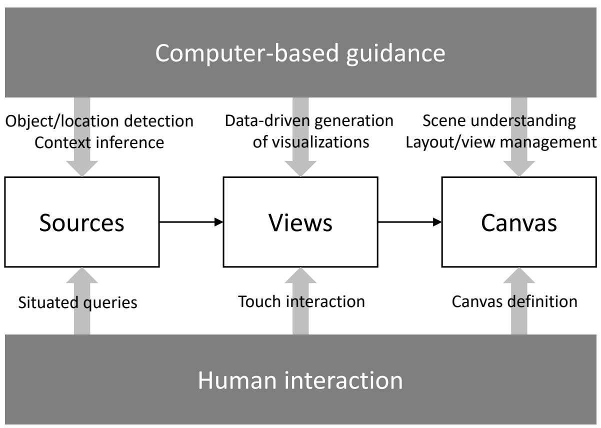

Figure 3: The main components of the proposed system architecture sources, views, and canvases. All components can be dynamically created and modified by either a human user or through automatic computer-based guidance.

Jetter et al. [12] argue that such a choice may not be necessary: Humans are capable of constructing conceptual blends [7], which integrate multiple familiar concepts into a new, emergent concept. Applied to embedded visualization, this means that users should be able to place conventional, abstract information visualizations in a registered canvas without overwhelming a user, provided the user is properly trained. Indeed, empirical comparisons of desktop visualization with AR visualization reveal that there are benefits to either approach, depending on the metric or objective [1].

Thus, the proposal of Jansen and Dragicevic [11] to use instrumental interaction for embedded visualization can be justified by its implied use of conceptual blending: Their instruments are either logical or physical tools, which can be mixed (blended) as needed. While physical tools are used for low-level interactions, such as pointing for specifying a range, logical tools are represented as dataflow pipelines. Such a conceptual distinction of physical and logical tools seems appropriate, because the power of dataflow primarily derives from its abstract, graph-based structure, which would likely have poor expressive leverage [17] in a physical embodiment. Manipulation of dataflow graphs using tangible interfaces interfaces [5] or 3D direct manipulation [6] easily leads to clutter and cumbersome interaction. However, conceptual blending should allow us to combine physical instruments for low-level tasks with logical instruments for dataflow authoring.

The VIGO (views, instruments, governors, objects) framework proposed by Klokmose and Beaudouin-Lafon [14] provides a suitable software architecture for implementing instrumental interaction. VIGO achieves this by extending the aforementioned MVC pattern: VIGO objects correspond to MVC models, but MVC controllers are subdivided into instruments and governors in VIGO. Instruments are made available to the user, while governors are responsible for ensuring semantic consistency and transformation. We shall rely on these ideas in the implementation of our framework.

With these considerations in mind, we can proceed to put forward ideas for a blended design of embedded visualization authoring.

DESIGN PROPOSAL

Inspired by the idea of modeling percepts and instrumental interaction, we propose a system design consisting of three key components: Canvases, views, and sources (Figure 3).

Canvases. Our system opportunistically turns appropriate surfaces at the task location into canvases, on which conventional 2D visualizations are shown. Placement on physical surfaces supports external cognition [15] and lends itself to "virtual" touch interaction assisted by passive haptic feedback [8, 20, 21].

Canvases can either be placed by the analyst or automatically determined. The latter requires a combination of multiple technologies: Appropriate placement on empty, flat surfaces requires sensor-based scene understanding, e.g., from depth cameras. A modified label placement algorithm selects suitable locations, while minimizing the distance to important referents, such as handles or other features of the machinery. Knowledge about the referents can be retrieved from a component that analyzes CAD models and cross-references it with those (potential) referents in which the analyst has expressed interest.

We believe that it suffices to define canvas as rectangular areas, which conveniently fit onto flat or slightly curved surfaces. If more sophisticated scene understanding is available [16], one can constrain canvas definition using geometric features of the scene, such as straight edges, vertical/horizontal alignment or enforced Manhattan-style placement in larger areas. We can also combine multiple connected (or even unconnected) rectangles to form a larger continuous canvas [13].

With a proper embedding of the canvas, such work can be perceptually integrated rather than piecewise and disjoint. For example, the analyst can make physical adjustments to machinery, while observing changes to telemetry data, such as heat or pressure, in an abstract view.

In the proposed software architecture, canvases assume the role of governors. They constrain the flow of information, while relieving instruments of their obligation to ensure consistency, thereby promoting them to universal instruments, which can mimic real-world behavior: For example, a pen could write on any canvas, irrespective of its content. A canvas governors would clean up any mess a pen leaves behind.

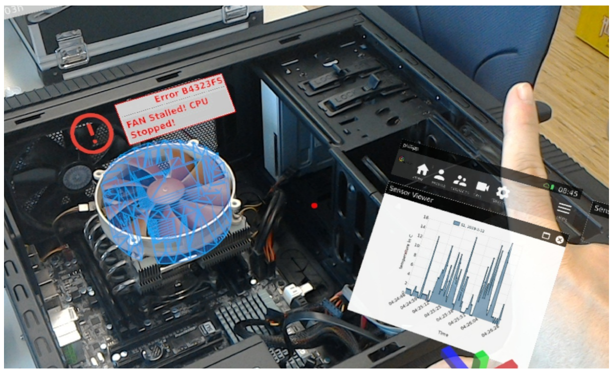

Figure 4: Seen through the Hololens: A registered CAD model showing the part of interest combined with an error notification (red) and real-time temperature data within the head-referenced User-Interface.

Views. are just conventional 2D visualizations, which makes it easy to repurpose existing visualization systems, for example, using web technology [18]. To be displayed, a view must be assigned to a canvas. This approach effectively circumvents the long-standing argument about the utility of 3D in infovis: The visualizations are visually *encoded* in 2D, but are *embedded* in 3D: They are situated in 3D with respect to a referent (provided the canvas has a referent) and registered in 3D with respect to a canvas.

Note that views in VIGO are described as "translations of the objects into one or more modalities perceptible by the user." In other words, views are device-dependent visual encodings, originally intended to support display ecologies and distributed user interfaces. However, we can easily repurpose such device-dependent views to express perceptually-dependent properties of views, i.e., to model the requirements imposed on a view by its canvas.

Sources. of data encoded in views are always accessed dynamically over a wireless network. While late binding of data to visualizations (i.e., the dynamic generation of visualizations) is established practice, it becomes crucial in dynamic environments. The analyst may not know the task location or specific referent in advance and must rely on dynamic queries to establish the link to the data source. For example, a fault may be reported in advance, but the actual physical cause may only become apparent when visiting the task location. Assuming the referent (the faulty component) can be identified by a spatial query (e.g., by 3D pointing), the analyst can bypass the cumbersome procedure of identifying the referent through conventional means (e.g., by reading a part number off a label and typing it into an app on a tablet).

Authoring and analysis as an interwoven dynamic process. The proposed system facilitates embedded authoring: Once arrived at a site, the analyst queries for currently visible or otherwise related referents, defines canvases, instantiates views, places views in canvases and connect views to data sources. Optionally, some or all of these tasks can be automated, relieving the analyst of manual setup. Moreover, we consider dynamically updating views from streaming sources, such as sensor data coming from a CPS. Data streaming is particularly important when the analyst's actions in the real world change the logical view and vice versa.

CONCLUSIONS AND FUTURE WORK

We admit that we have just started to build a prototype implementation of our design for the Microsoft Hololens. By the time of the workshop, we are confident that we will be able to report on first experiences with this prototype. It will be focused on building simple visualization pipelines with standard encodings (graphs, scatterplots, tables etc.), but will incorporate live referent discovery using place recognition as well as real-time data streaming from online (MQTT[1]) sources.

While we feel these capabilities will already be useful and novel, many important aspects have not even been considered yet. These include, among others, deep pipelines with dataflow through multiple stages, multi-device interaction and multi-user collaboration. Nonetheless, we are confident that the power of embedded visualization authoring will be clearly apparent from our work.

[1] http://mqtt.org/

REFERENCES

[1] Benjamin Bach, Ronell Sicat, Johanna Beyer, Maxime Cordeil, and Hanspeter Pfister. 2018. The Hologram in My Hand: How Effective is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented Reality? *IEEE Transactions on Visualization and Computer Graphics* (2018). https://doi.org/10.1109/TVCG.2017.2745941

[2] Benjamin Bach, Ronell Sicat, Hanspeter Pfister, and Aaron Quigley. 2017. Drawing into the AR-CANVAS: Designing Embedded Visualizations for Augmented Reality. In *Proc. Immersive Analytics Workshop*.

[3] Michel Beaudouin-Lafon. 2000. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 446–453. https://doi.org/10.1145/332040.332473

[4] Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View management for virtual and augmented reality. In *UIST (User Interface Software and Technology): Proceedings of the ACM Symposium*. https://doi.org/10.1145/502360.502363

[5] Mark Billinghurst, Raphaël D. Grasset, and Hartmut Seichter. 2010. Tangible Interfaces for Ambient Augmented Reality Applications. In *Human-Centric Interfaces for Ambient Intelligence*. https://doi.org/10.1016/B978-0-12-374708-2.00011-5

[6] Barrett Ens, Fraser Anderson, Tovi Grossman, Michelle Annett, Pourang Irani, and George Fitzmaurice. 2017. Ivy: Exploring Spatially Situated Visual Programming for Authoring and Understanding Intelligent Environments. In *Proceedings of the 43rd Graphics Interface Conference* (Edmonton, Alberta, Canada) *(GI '17)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 156–162.

[7] Gilles Fauconnier and Mark Turner. 1998. Conceptual integration networks. *Cognitive Science* (1998). https://doi.org/10.1207/s15516709cog2202_1

[8] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011*, Jeffrey S. Pierce, Maneesh Agrawala, and Scott R. Klemmer (Eds.). ACM, 441–450. https://doi.org/10.1145/2047196.2047255

[9] Robert J.K. Jacob, Orit Shaer, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-Based interaction: A framework for post-WIMP interfaces. In *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/1357054.1357089

[10] Yvonne Jansen and Pierre Dragicevic. 2013. An interaction model for visualizations beyond the desktop. *IEEE Transactions on Visualization and Computer Graphics* (2013). https://doi.org/10.1109/TVCG.2013.134

[11] Yvonne Jansen, Pierre Dragicevic, and Jean Daniel Fekete. 2013. Evaluating the efficiency of physical visualizations. In *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/2470654.2481359

[12] Hans-Christian Christian Jetter, Harald Reiterer, and Florian Geyer. 2014. Blended Interaction: understanding natural human–computer interaction in post-WIMP interactive spaces. , 1139–1158 pages. https://doi.org/10.1007/s00779-013-0725-4

[13] Brad Johanson, Greg Hutchins, Terry Winograd, and Maureen Stone. 2002. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France) *(UIST '02)*. Association for Computing Machinery, New York, NY, USA, 227–234. https://doi.org/10.1145/571985.572019

[14] Clemens Nylandsted Klokmose, Dk Arhus, and Michel Beaudouin-lafon. 2009. VIGO : Instrumental Interaction in Multi-Surface Environments. *Architecture* (2009), 869–878. https://doi.org/10.1145/1518701.1518833

[15] Zhicheng Liu, Nancy J. Nersessian, and John T. Stasko. 2008. Distributed Cognition as a Theoretical Framework for Information Visualization. *IEEE Trans. Vis. Comput. Graph.* 14, 6 (2008), 1173–1180. https://doi.org/10.1109/TVCG.2008.121

[16] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D. Wilson. 2016. SnapToReality: Aligning Augmented Reality to the Real World. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris, and Juan Pablo Hourcade (Eds.). ACM, 1233–1244. https://doi.org/10.1145/2858036.2858250

[17] Dan R. Olsen. 2007. Evaluating User Interface Systems Research. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, Rhode Island, USA) *(UIST '07)*. Association for Computing Machinery, New York, NY, USA, 251–258. https://doi.org/10.1145/1294211.1294256

[18] Gheric Speiginer and Blair MacIntyre. 2019. A Practical Approach to Integrating Live 2D Web Content with the Immersive Web. In *The 24th International Conference on 3D Web Technology, Web3D, Los Angeles, California, USA, July 26-28, 2019*, Nicholas F. Polys, Mike McCann, Feng Liu, and Andreas Plesch (Eds.). ACM, 1–10. https://doi.org/10.1145/3329714.3338136

[19] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. 2017. Embedded Data Representations. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (jan 2017), 461–470. https://doi.org/10.1109/TVCG.2016.2598608

[20] Robert Xiao, Chris Harrison, and Scott E. Hudson. 2013. WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013, Extended Abstracts*, Wendy E. Mackay, Stephen A. Brewster, and Susanne Bødker (Eds.). ACM, 2889–2890. https://doi.org/10.1145/2468356.2479563

[21] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Trans. Vis. Comput. Graph.* 24, 4 (2018), 1653–1660. https://doi.org/10.1109/TVCG.2018.2794222