A Component-Based Approach towards Mobile Distributed and Collaborative PTAM

Tim Verbelen[*] Pieter Simoens[†] Filip De Turck[‡] Bart Dhoedt[§]

Ghent University - IBBT, Department of Information Technology
[†]Ghent University College, Department INWE

Abstract

Having numerous sensors on-board, smartphones have rapidly become a very attractive platform for augmented reality applications. Although the computational resources of mobile devices grow, they still cannot match commonly available desktop hardware, which results in downscaled versions of well known computer vision techniques that sacrifice accuracy for speed. We propose a component-based approach towards mobile augmented reality applications, where components can be configured and distributed at runtime, resulting in a performance increase by offloading CPU intensive tasks to a server in the network. By sharing distributed components between multiple users, collaborative AR applications can easily be developed. In this poster, we present a component-based implementation of the Parallel Tracking And Mapping (PTAM) algorithm, enabling to distribute components to achieve a mobile, distributed version of the original PTAM algorithm, as well as a collaborative scenario.

1 Introduction

Component-based development is not a new approach towards augmented reality (AR). Already in the early 2000s the need grew to create AR frameworks that offer components providing common functionality such as marker tracking, scene graph management and rendering [1], which enables developers to quickly prototype and create new AR applications on top of such a framework. By distributing some of these framework components (i.e. a distributed scene graph), this evolved to collaborative AR [5], where multiple users can see and interact with the same virtual objects.

When mobile devices such as personal digital assistants (PDAs) became available, AR research shifted towards handheld AR, and mobile AR frameworks were proposed [6]. To increase the tracking performance, the tracking process was executed on a server connected to the mobile device via WiFi [8].

Nowadays, smartphones are becoming more and more the preferred platform for mobile AR, as they are equipped with cameras, GPS sensor, accelerometers. Although their computational resources are not up to par with common desktop or server infrastructure, they do have enough processing power to allow real-time detection and tracking for augmented reality [7]. However, to cope with the limited processing power on the mobile device, these algorithms are often heavily modified versions of known state-of-the-art algorithms, sacrificing accuracy and quality to gain speed [3].

Instead of deciding at design time which parts of the AR application should run on the mobile device and which should run on a remote server, we propose to adopt a component-based approach that decouples software components from their execution location. These components are configured and distributed at runtime, which allows the application to adapt for different heterogeneous mobile devices. Component-based development also makes it easy to extend applications by adding components, enables efficient reuse of existing components, and transparently allows collaborative applications by sharing components between different users.

In this poster, we present a component-based implementation of the Parallel Tracking And Mapping (PTAM) algorithm [2], a widely used tracking algorithm in a priori unknown environments. In comparison with PTAMM-Plus [4], our components not only provide better modularity, but they can also be configured and distributed at runtime. Using a component framework implemented in Java, the components can run on Android, the most popular mobile OS, as well as on commodity hardware such as desktops or servers. We illustrate the main advantages of our component-based approach.

2 Component-based PTAM

Using a representative AR application based on the PTAM algorithm, we identified the following components as shown in Figure 1:

VideoSource The VideoSource fetches video frames from the camera hardware. These frames are analyzed by the Tracker, and rendered with an augmented reality overlay by the Renderer.

Renderer Each camera frame is rendered on screen together with an overlay of 3D objects. These 3D objects are provided by the Model, and aligned according to the camera pose given by the Tracker.

Controller The Controller handles user input events originating from the screen (Renderer) and translates these to actions to modify the Model.

Model The Model manages all 3D objects and their state.

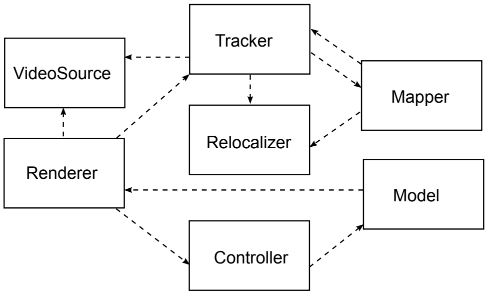

Figure 1: A component-based approach on PTAM: components are provided for capturing videoframes (VideoSource), rendering 3D objects (Renderer) and interacting with 3D objects (Controller and Model). The tracking process in captured in the Tracker component, using the Relocalizer component when tracking is lost. The Mapper component initializes, extends and refines the map.

[*]e-mail:tim.verbelen@intec.ugent.be
[†]e-mail:pieter.simoens@intec.ugent.be
[‡]e-mail:filip.deturck@intec.ugent.be
[§]e-mail:bart.dhoedt@intec.ugent.be

Tracker The Tracker searches map points in video frames and calculates the new camera pose. When a new keyframe should be added to the map, this keyframe is sent to the Mapper.

Relocalizer When not enough map points are found in the video frame, the Tracker calls the Relocalizer, that estimates the camera position using the same approach as described in [2].

Mapper The Mapper receives keyframes from the Tracker, that are used to initialize and extend the map. The mapper also performs the bundle adjustment to optimize the map points. Other components such as the Tracker and Relocalizer register themselves as MapListeners, that receive notifications when the map is updated.

These components are implemented based on the source code of [2]. The resulting application is shown in Figure 2. On the right a grayscale video frame is shown with the tracked feature points, from which the camera position is estimated. The left shows the resulting overlay with a 3D object.

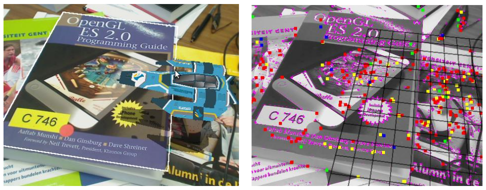

Figure 2: The augmented reality application tracks feature points in the video frames (right) to enable the overlay of 3D objects (left).

3 Mobile, distributed PTAM

Instead of limiting the tracking to the coarse images for tracking on the mobile phone, as proposed by Klein et al. [3], the Tracker component is made configurable to tailor the Tracker to the available CPU resources for each device. In the fine tracking stage, the number of points reprojected and sought in the image can be limited to a lower threshold (1000 in the desktop version). Also, the FAST feature detection stage can be omitted, in which case the features are matched using zero-normalized SSD score of a 8x8 template in a 5-pixel-radius. Another important factor on the tracking performance is the resolution of the input images, which can be configured in the VideoSource component. By tuning these parameters, the application is tailored for each device at runtime. For example, if one desires a tracking frame rate of 15 frames per second, on the LG Optimus 2x an input resolution of 800x480 and a search for 400 points suffices, while on the HTC Desire the input resolution should be limited to 400x240 and only 200 points can be reprojected to meet the required performance.

To mitigate the high computational cost of the mapping process, and in particular the global bundle adjustment, while still allowing map sizes comparable to the PC version, we can offload the Mapper component to a PC or laptop connected to the mobile device via a wireless connection. Because the mapping is a background process, the delay of sending a camera frame to the server has little impact on the user experience, while this speeds up the global bundle adjustment. On the LG Optimus 2x for example, offloading the Mapper to a laptop results in a speedup of a factor 10.

4 Collaborative PTAM

Another application of the component-based AR application, is collaborative AR by sharing components. In this scenario, the Model and Mapper components are shared between multiple users. This way, all users get the same map updates, and can track the same environment. Also, all users can add keyframes to the map, which can be built and refined in a collaborative way. By offloading and sharing the Model component, multiple users can interact with the 3D objects.

To get such a setup, the Mapper component is deployed on a laptop connected to multiple mobile devices via WiFi. All users have a device running a VideoSource, Renderer, Controller, Tracker and Relocalizer component, and connect to the laptop running the Mapper and Model components as shown in Figure 3.

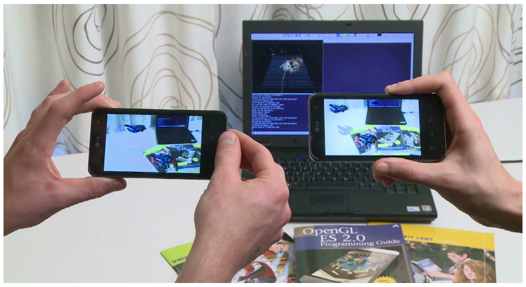

Figure 3: Collaborative PTAM on two mobile devices: the map is updated and refined on the laptop with keyframes from the two devices. Both devices track the map and render the shared scene graph.

5 Conclusion

In this poster we presented a component-based middleware platform that configures and distributes software components at runtime. We showed the effectiveness of this approach for augmented reality applications, using a component-based implementation of the PTAM algorithm. By configuring the components at runtime and if possible offloading components to a server connected via WiFi, performance can be increased and the application can be tailored for different mobile devices. By sharing components between multiple users, a collaborative PTAM algorithm is achieved.

Acknowledgements

Tim Verbelen is funded by Ph.D grant of the Fund for Scientific Research, Flanders (FWO-V).

References

[1] M. Bauer, B. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, and M. Wagner. Design of a component-based augmented reality framework. In *Proceedings of the International Symposium on Augmented Reality (ISAR'01)*, ISAR '01, 2001.

[2] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 6th International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, 2007.

[3] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Proceedings of the 8th International Symposium on Mixed and Augmented Reality*, ISMAR '09, pages 83–86, 2009.

[4] T. Nguyen, C. Sandor, and J. J. Park. PTAMM-Plus: Refactoring and extending PTAMM. In *In Proceedings of International Conference on Artificial Reality and Telexistence*, pages 84–88, 2010.

[5] G. Reitmayr and D. Schmalstieg. Mobile collaborative augmented reality. In *Proceedings of the International Symposium on Augmented Reality (ISAR'01)*, ISAR '01, 2001.

[6] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Proceedings of the Third international conference on Pervasive Computing*, pages 208–219, 2005.

[7] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):355 –368, 2010.

[8] D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. In *Proceedings of the 7th International Symposium on Wearable Computers*, ISWC '03, 2003.