ARTool - Augmented Reality Human-Machine Interface for Machining Setup and Maintenance

Amedeo Setti, Paolo Bosetti, and Matteo Ragni

Abstract In modern production lines, smaller batches to be produced and higher customization level of a single component bring to higher cost, related especially to setup and preparation of machines. The setup of a milling machine is an operation that requires time and may bring to errors that can be catastrophic. In this Chapter, the **AR**Tool Augmented Reality framework for machine tool operations is presented. The framework permits to write and debug part-code in an augmented environment, to identify quicker misalignments and errors in fixing of new blank

Amedeo Setti

Pro-M Facility, Trentino Sviluppo S.p.A., *Rovereto (TN), Italy*, e-mail: `amedeo.setti@trentinosviluppo.it`

Paolo Bosetti

Department of Industrial Engineering, University of Trento, *Trento, Italy*, e-mail: `paolo.bosetti@unitn.it`

Matteo Ragni

Department of Industrial Engineering, University of Trento, *Trento, Italy*, e-mail: `matteo.ragni@unitn.it`

material, and to support maintenance operations. The ego-localization of the hand-held device that depicts the augmented scene in machine work-area is based upon markers. The library that performs marker identification is brand-new and it is benchmarked throughout the Chapter against a state-of-the-art solution (ARUCO) and a ground truth (multi-stereoscopic motion capture). The Chapter also describes the general information flow and the context that brought to the conception of the **AR**Tool framework, and presents a series of applications developed using the framework.

1 Machining Economics and Augmented Reality

The economics of machining operations considers different cost authorities that should be minimized to achieve an efficient process. For each machined product, the main factors to consider are [13, 17]:

- the cost of the effective machining operation, alongside with maintenance and man-hours costs;
- the cost for preparing the machine, which comprises testing of the part-program, fixing and aligning the blank material in the working area, and mounting the tools and the cutters on tool holders;
- the costs for loading the raw material and unloading the finished part;
- the cost of tooling.

One of the cost of greater impact is due to maintenance and inactivity that directly correlates time and machining costs. In case of human operator involved in the process of loading and unloading material—e.g. in case case of shop-floor with limited automation and with small batches to be produced—optimizing the maintenance and the alignments procedures permits to reduce dramatically the costs.

The Chapter describes a framework that exploits Virtual and Augmented Reality technologies to reduce unproductive times. The platform, namely **AR**Tool, reduces errors induced by operators during procedures such as alignments of blank material. In common practice, for avoiding collisions that may result in extended damages for both machine and work-piece, in-air test are performed—i.e. a complete execution of the part program with a constant safety offset between the tool and the raw material.

The Augmented Reality (AR) component of the **AR**Tool frameworks uses the reference systems stored inside the machine controller to overlaid a properly oriented simulation of the workpiece blank, alongside with fixtures, and machine moving peripherals on the scene of the working area captured by a camera. The simulation reflects exactly what the machine is programmed to perform, thus in-air test, which may require hours to be fully executed, is substituted by an augmented simulation with time scaled. The operator concentrates the attention only on the complicated passages, and effectively identify visually evident mistakes, in less time and with an higher accuracy.

The augmented component of the framework is built to run on a personal device, and throughout the Chapter the considered device is a tablet which is relatively low cost with respect to more exotic hardware—e.g. head mounted displays. With a tablet, the operator explores the simulated scene from different perspectives. Moreover, the same framework can be easily employed to enhance the maintenance operations on a machine, and inexperienced operators largely benefit from the usage of augmented schematics and manuals.

1.1 Envisioning AR Technologies

The manufacturing industry has always envisioned the application of AR related technologies, and the strong interest is underlined in the results of the survey conducted by the *Deutsche Forschungszentrum für Künstliche Intelligenz* during the Hannover Messe of 2010. On a total of 54 industrial rappresentative, the 77.8 % have every intention of deploying augmented solutions in their production lines [28].

In literature, Architecture is the first field that embraced the AR, enlarging the Building Information Modeling schemes in order to accomodate a data infrastructure for the Augmented Reality technologies [35].

Also the Cognitive Sciences inspected the application of AR technologies, evaluating the benefit from a cognitive workload point of view [15, 30].

1.2 Manufacturing and Augmented Reality

In general, the proofs-of-benefit for AR as alternative training method, described in [12, 27], make educational and informational applications, such as augmented manuals and operators training, literally mainstream. In [21], the authors use a marker solution to build interactive lectures on machinery handling for completely inexperienced students, revealing once again the high acceptance of the methodology, and allowing a faster comprehension of programming caveats for complex paths [7]. [32] pushes towards the integration of AR for training and expert systems to support decision making for inexperienced operators.

The costs of integrating such a new technology in the process is not an easy decision. Few studies started to develop decision supporting tools *ex-ante* [9], for evaluating the effectiveness of the approach for a specific manufacturing process. Both Product Design and Planning (PDP) and Workplace Design and Planning (WDP) benefit from an AR developing environment [24], that aid designers and engineers in making better decision while designing new assembly lines. Lines include AR interfaces [3, 5] that guide the operator in the execution of a specific task — i.e. projecting welding spots on work-piece in [8]. The ergonomy of the technology is also evaluated in literature [34].

Papers [15, 36] present first implementations of virtual assembly interfaces. Cameras are used to detect position of operator hands, that are the Human Computer Interface (HCI) for the augmented renderer. Systems are desktop static prototypes, but usability is validated with respect to non-augmented real-case-scenario. Evidence of cognitive workload reduction for the operator are underlined, as also reduced time to complete tasks and reduced mean error rate.

For what concerns application on process machines, the manipulators programming and collision avoidance is for sure the most prolific field. And in fact the complex kinematic configurations during a program execution results more intuitive — e.g. programming [10] or visualizing [6, 11] end-effector pose and trajectory, — by the mean of different user interface — e.g. mobile, projection on half silvered glasses or head displays [16]. General survey can be found in [25] and [22].

The applications of AR on machine tools are limited and may be referred as proof-of-concept prototypes rather than proof-of-benefit ones. In [31], an AR application is used to help operators during manual alignment in a pipe manufacturing machine. In [20], AR is used to develop a framework for dimensional validation of finished parts. The framework is marker based, one of the more reliable solution that guarantees enough precision for manufacturing applications. The works also illustrates evidences of advantages, both economical and practical, induced by the use of AR applications in manufacturing. Another approach typically discussed in literature, is the use of super-imposition of virtual image on work-space video recording for validation of complex paths [37]. Virtual images contain augmented information about the process, and are visualized through the use of different device, such as stereo-projector [23] or mobile devices. In general, the idea is to use the augmented visualization to give more insight to the operators about the process, usually before performing the actual machining operation [39]. Other applications focused instead on active maintenance, using OCR (Optical Character Recognition) in combination with localization markers [19], but real benefits of such implementations to users were not assessed. In [38], it is worth noting the use of handheld devices, with respect to the typical static desktop setups seen in previous works.

1.3 Chapter summary

The complete concept for the full **AR**Tool framework is deeply analyzed in Section 2: starting from a broader view, the single elements of the approach are described and motivated. The device layer description is the pretext to introduce the `ARSceneDetector` , in Section 3.2: the library is portrayed extensively and benchmarked against to the state-of-the-art equivalent library `ARUCO`, and results are illustrated. Section 4 is an application showcase that presents some of the developed applications that use **AR**Tool for data interchange.

2 The ARTool Platform

The **AR**Tool Framework is conceived to support machine manufacturers, technical offices, and machine operators in bringing augmented reality information on the machine and in the production lines.

The main objective of the framework is the optimization of the machining processes by tackling two major shortfalls:

- reducing the unproductive time between production batches, allowing the operators to test quickly the newer part-program and eventually correct misalignment of blank material with respect to to reference systems saved in numerical control;
- supporting the maintenance procedures through augmented manuals that facilitate remote assistance from technical support. Failure diagnosis can be improved highlighting failing components directly on machine chassis.

2.1 From Authors to Consumers: the Flow of Data

The main source of information are the technical offices, that provide tasks to shop-floor. Tasks data include:

- part-programs;
- fixtures list and fixture sequences;
- tooling information.

The technical office stores the authored data in SCADA (Supervisory Control and Data Acquisition) servers: this permits the centralization and distribution to data consumers.

The second authoring agent of the network is the machine manufacturer that through a Content Delivery Network distributes assets for augmented manuals that the different SCADA servers of the different industries that acquired the machine download.

The SCADA server act as a gateway for delivering update data to local machine and shop-floor operators.

For both technical offices and machine manufacturers, tools for authoring information are developed as plugins for commercially available Computer Aided Engineering (CAE) software [33, 26]. For technical offices, this means expand the capabilities of common Computer Aided Manufacturing software, while, for manufacturers, the plugins are related to Computer Aided Design (CAD) and Product Life-cycle Management (PLM) software. Optionally, manufacturers can exploit the framework for marketing opportunities, such as ticketing services and web store for spare parts.

The main information consumer are the machine tool and the operator device. Both consumer download data from SCADA servers. The computer numerical control (CNC) communicates using a client that can be software service, for newer machines, or a embedded computer, for older machines. The client requires an implementation of the proprietary communication protocol of the machine, while the communication with the SCADA is performed through standard protocols. The client broadcasts to the SCADA server all relevant information for diagnostic and simulation purposes, such as system states, tools table, etc.

Machine operators carry a personal device that has the hardware necessary to perform the ego-localization task—i.e. camera and inertial sensors—that is the most prominent feature of the **AR**Tool framework. Currently, **AR**Tool has been tested only on tablet devices, which are relatively low-cost and reliable with respect to other solutions.

2.2 Operator device

Operators are equipped with personal devices that have the minimum hardware requirements to perform the ego-localization. The current release of **AR**Tool framework requires an high definition camera for gathering the scene on which assets are overlaid, an inertial measurement unit to filter the ego-localization state and a GPU for rendering the virtual scene.

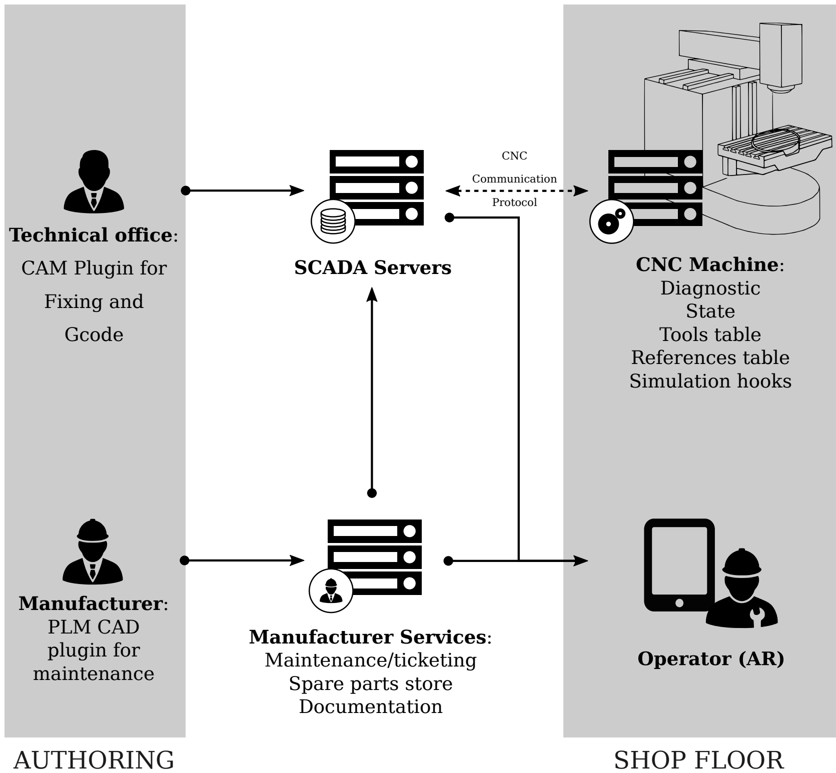


Fig. 1 The **AR**Toolflow of information, from technical offices and manufacturer, to machines and operators

Localization is performed through markers that characterize a scene (cfr. section 3.2 for a description of *scene* in detail). Once a scene is identified, a query to the SCADA server permits to populate the camera feed with virtual assets.

The framework eases the presentation of different information, that are contextualized with respect to a scene and a *operation mode*, or scope. When the current scope is to setup a new process for a machine, the main assets considered are:

- blank material and possibly the fixing for the bulk;

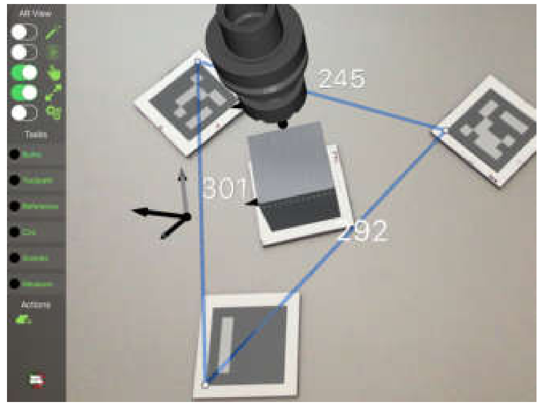


Fig. 2 Screenshot of a very first prototype **AR**Tool iPad app, showing the setup-mode augmented reality view. In this case, marker distances are measured. Camera images are localized in the working area: the application shows a bulk, a trajectory and a tool oriented with machine reference frames

- tool and optionally machine head;
- mechanical axes simulacra;
- coordinate systems and oriented trajectory;
- marker anchoring elements (cfr. section 2.4);
- auxiliary descriptive text.

When the intended scope is maintenance, the framework is designed to stage:

- machine contextual information;
- mask for component of the machine;
- contextual manual web pages;
- geometric primitive shapes—e.g. arrows—that can be used to draw operator attention.

The device selected as prototype is an Apple iPad 2 Air Tablet, with iOS 9.3 operating system. The framework is a `C++` library that exposes `Swift` and `Objective-C` bindings. The rendering operation are handled by the Apple Framework `SceneKit` [1].

2.3 The SCADA and per-machine server

The SCADA server is responsible for storage and distribution of augmented assets. It also challenges machine clients for information necessary to present simulation and localized elements:

- the current state of machine, that includes the current position of axes, the active coordinate system, the loaded tool on the spindle and the active part-program;
- part-program simulation hooks, that comes from the numerical control parser/interpolator. If this information is available, **AR**Tool shows the exact tool trajectory as interpolated by the numerical controller. If this information is not actually available, the framework exposes a fallback interpolator, that will generates trajectory with minimal differences;
- coordinate systems table and tool table. The tool table relates the currently loaded tool with a solid model counterpart for rendering. The reference systems table permits to project machine simulacra within the AR view, alongside the correct origins;
- optionally, diagnostic information that guides inexperienced operators in unusual situation and training.

In the experimental system, the server is a `Ruby` and `C++` software on a separated machine, with database composed by a sequence of `YAML` files—i.e. a format that simplify inspection and debugging. The server provides a `HTML5` web application for authoring, which exploits the `C++` component of **AR**Tool framework for the creation of scenes from static images.

2.4 ARTool as input

Capabilities of **AR**Tool can be enriched from a simple output interface, to a novel, input/output human machine interface, providing functionalities for identifying exterior points and geometric features in space.

The screen of a mobile device can be used to capture a bi-dimensional input. As already discussed, depth can be reliably reconstructed by using structured elements (markers). Each marker defines a *virtual plane*. Indeed, the area of the screen can be projected on this plane, associating each bi-dimensional screen coordinate to a

Fig. 3 An example application created using **AR**Tool framework, that helps operator in bulk alignment operations. This application is described in Section 4.4

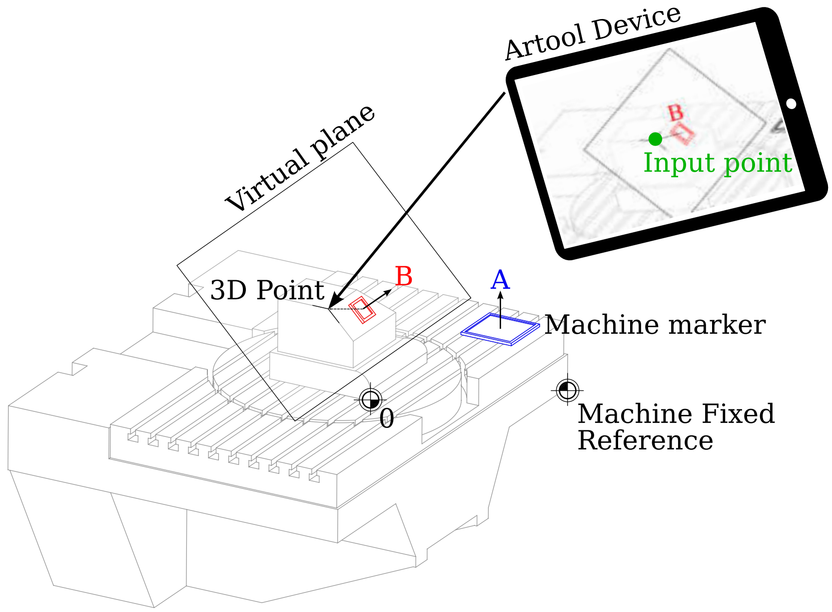


Fig. 4 Using the mobile device as a 3D input system, through a mobile marker

tri-dimensional point that lies on the virtual plane. In other words, that point is the projection of the 2-D point on screen along the line of view on the virtual plane.

The procedure is explained in Fig. 4. Each machine has a *fixed* origin, which is hard-coded in machine's controller. Then, the machine may define an *active* reference frame (in this case **0**), that is used for defining the coordinates in the part-program. The transformation matrix from *fixed* to *active* reference frame is known. In the figure, reference system **A** is defined by a *machine marker*, whose position is well known with respect to the machine *fixed* origin. Through a simple coordinate transformation, the vector from reference **0** to reference **A** is known. The marker in **A** is used by **AR**Tool library to ego-localize the mobile device, so that the vector from **0** to tablet internal reference is known. **AR**Tool also reconstructs the vector pointing to the marker reference **B**, which is the movable virtual plane, closing the chain between **0** and **B**. When the user taps the mobile screen, the 2D coordinates of the tap on screen are transformed in the coordinates of a 3D point projected on the plane of **B**.

There is no need to keep both *machine marker* and *moving marker* framed at all times: indeed, once the position of the free marker is set, it can be anchored in software while framing both, then anchoring allows **AR**Tool to use the *free marker* as a *machine marker*, thus ego-localizing the device relatively to any marker in the markers chain. This opens to the possibility to create *chains* of markers, altough the reliability of the ego-localization decreases exponentially at each hop.

Fig. 3 shows a practical application. In common practice, part-programs contain axes motion coordinates relative to a point in space, which is the *workpiece origin*. One of the very first operations is to identify the position of *workpiece origin* on the workpiece in the working space. This requires to approach the object with a touching probe—i.e. the tool in the figure—that returns a feedback to the machine controller upon achieving contact. The **AR**Tool application acts as a virtual touching probe, that identifies a point that lies on the virtual plane described by the marker attached on the workpiece. In the figure, the identified point is the upper-left corner of the gray cube overlaid on the workpiece, visible on the screen of the tablet.

3 `ARSceneDetector` : the Core of ARTool

One of the critical requirements for an augmented application is a reliable and precise localization of the device with respect to the scene observed. The library `ARSceneDetector` is the software component that fulfill this task.

During the early development stage, the **AR**Tool framework included the open source libray `ARUCO`, currently distributed with the OpenCV suite [4]. `ARUCO` is a localization library which takes advangtage of the presence of structured markers in scene for reconstruction. `ARUCO` was chosen after a comparison with the ArtoolKit platform: it provided a better responsiveness at the cost of a lower accuracy, on the prototype device.

In a later development stage, in order to tackle the accuracy issues and to get a more stable and reliable localization through sensor fusion, the designed from scratch `ARSceneDetector` library has been introduced as core component of the **AR**Tool framework. The library is strongly device dependent (ARM-processor)and uses specific hardware instructions to speed-up its performances. This allows to squeeze the computational power of the device, attaining a precise and yet responsive placement of virtual assets on the framed scene.

The next section describes the internal logical structure od `ARSceneDetector` , while the section 3.2 is devoted to a comparison against `ARUCO`. In particular, ego-localization accuracy and computational efficiency are evaluated carefully.

3.1 `ARSceneDetector` Library Details

The `ARSceneDetector` library is logically divided in three different layers, from perception to scene rendering.

- The Sensor Acquisition and information gathering layer is written in `Swift` language. This is required by the platform and uses the current operating system API.
- The Marker Handler layer is written in `C++` and is linked to the OpenCV library. This layer handles the identification of the marker in the scene, the inter-frame tracking and the image stabilization.
- The very last layer is the Scene Detector, a classifier that extract more information based upon the relative position of the marker in the scene.

The three layers are presented in Fig. 5.

As with other computer vision algorithms, `ARSceneDetector` requires a calibration of the camera [14] which results in a camera matrix. Light parameters and thresholds are automatically evaluated through normalization procedures: each frame is enhanced and the edge detection is extracted from the frame in GPU.

Using the internal camera of the prototype device, it is possible to collect frame with 720p and 1080p resolution. The bigger the frame, the lower the update frequency guaranteed for the localization—i.e. 120 Hz and 30 Hz respectively.

Beside the camera frame, accelerations and angular ratios of the device are measured by the on-board IMU sensor. This information permits to stabilize the ren-

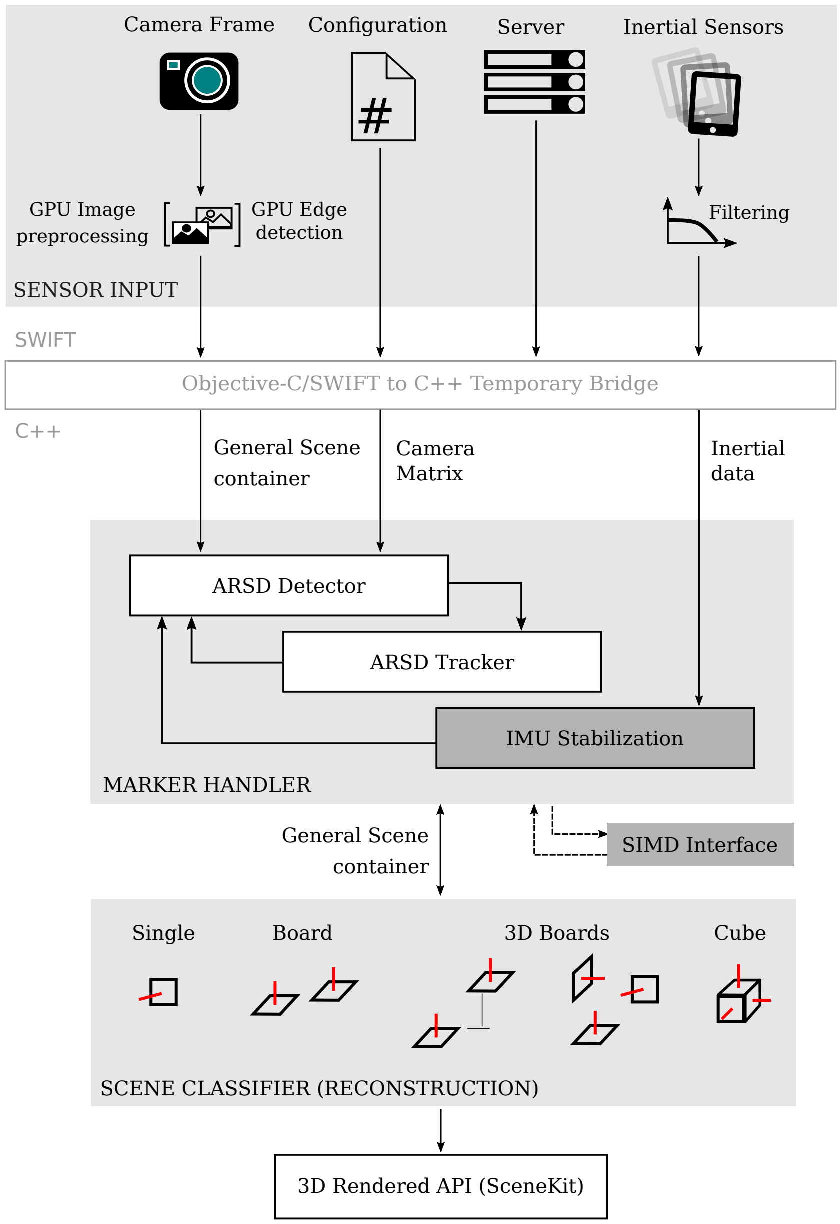


Fig. 5 Library structure. ARSD stands for `ARSceneDetector` . In gray, plugins that are disabled during benchmark

dered scene [2]. The combination of the frame and IMU data are passed through the bridge `Swift/C++` and enters the marker handler layer, as Scene container.

The Marker Detector is the implementation od a classical one-frame-at-the-time algorithm which, for each camera frame, extracts convex quadrilateral shapes as marker candidates. The candidate are then reoriented and checked for squareness. The pose of each square element is reconstructed using different well-known algorithms [18, 29]. The algorithms return a reference system that is oriented through an asymmetrical pattern drawn on the marker itself. The pattern can be a number encoded in a binary form—e.g. the `ARUCO` encoding—or a image. The reference system is relative to the camera point-of-view and has always the $\hat{z}$ axis perpendicular to the marker surface.

The Tracker is an extension of the Detector algorithm that uses information of the previous frame to reduce the computational efforts of the Detector, limiting the area in which quadrilateral are searched, and lowering the frequency of whole-frame scanning (configurable, but with a default value of 10 frame). It can be disabled. To improve efficiency, Single Instruction Multiple Data (SIMD) instructions are employed.

The IMU Stabilization block filters the state of the device, fusing the signal sampled by the IMU sensor.

The result of the Marker Handler is a General Scene Container, a data structure with all the information about identified marker and their position with respect to the device.

The very last layer of the library performs a classification of the General Scene Container. Using a combination of markers it is possible to drastically improve the accuracy of the localization. The possible scenes contain:

- a simple single marker;
- a board of co-planar markers, with parallel $\hat{z}$ axes;
- a board of markers, with parallel $\hat{z}$ axes, and known, non-zero offset in $\hat{z}$ direction;
- a board of three markers with mutually orthogonal $\hat{z}$ axes, with known offset vectors;
- a solid cube of markers.

The SCADA server provides the list of scenes to be classified. The Scene Detector matches the most similar one. Nevertheless, the library may enrich SCADA definitions: this particular feature is used for marker chaining which consents to expand the rendering volume, reaching area in which marker are not currently visible. Once the scene has been classified and reconstructed, the General Scene Container is shared with the render engine, that places the virtual models in a virtual world that is aligned with the perceived one.

3.2 Library benchmarking

This section is devoted to the comparison between the `ARSceneDetector` and the `ARUCO` library, which is the first solution adopted by **AR**Tool, in the very early developing stages.

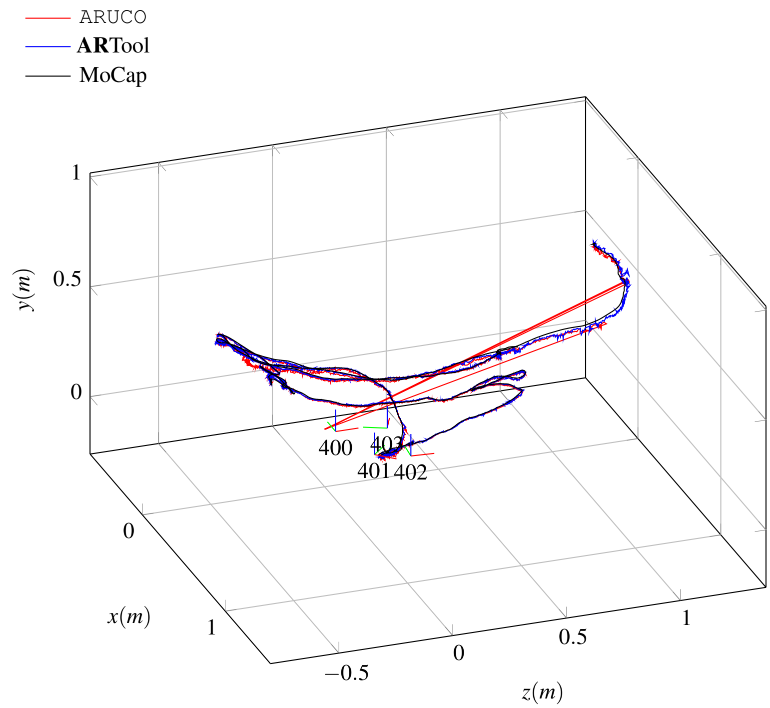
The test focuses on:

- computational time;
- reliability in marker identification;
- accuracy in ego-localization.

3.2.1 Methodology

For the localization, the ground-truth is provided by a professional level Motion Capture System (MoCAP - *OptiTrack*, equipped with 8 *Prime13* cameras running at 120 fps). For the localization test, a MoCAP 3D reference is attached on the iPad that records a video of a board of 4 `ARUCO` markers. At least one marker is always framed during the video (see Fig. 6). The MoCAP reference frame is placed on the coordinate system of one of the corner of one of the marker—i.e. the origin have a known offset.

The recorded video is than used to run a testing application with both libraries in profiling mode. Setup parameters are fine tuned to crunch the maximum performances without compromising too much reliability, but some of the very advanced feature of the `ARSceneDetector` —i.e. the GPU usage and the SIMD operations—are disabled for a fairer comparison. This effects the real performances of `ARSceneDetector` , but allows to limit the comparison only on the algorithmic level, rather than on differences in filtering and input data processing.



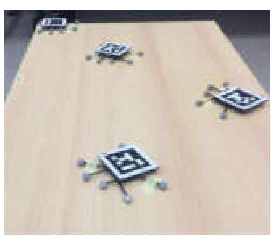


Fig. 6 A frame of the video used for bench-marking

Since the signal length are different for MoCAP and iPad, localization data are synchronized minimizing the variance of positions with respect to time. Given the signals:

- $x_0(t)$ the x coordinate returned by the motion capture at frame t
- $x_A(t)$ the x coordinate returned by the **AR**Tool library at frame t
- $x_B(t)$ the x coordinate returned by the ARUCO library at frame t

the distance $\varepsilon_x(t,\delta)$ is evaluated as:

$$\varepsilon_x(t,\delta) = 2x_0(t) - ((x_A(t+\delta) + x_B(t+\delta)) \tag{1}$$

while the variance $\sigma_x(\delta)$ with respect to the shift δ on the x signal is obtained as:

$$\sigma_x(\delta) = E\left[\varepsilon_x(t,\delta) - E\left[\varepsilon_x(t,\delta)\right]\right] \tag{2}$$

consequently, the time-shift to be used for aligning the signals is the result of:

$$\delta^* = \arg\min_{\delta} \sum_{i=\{x,y,z\}} \sigma_i(\delta) \tag{3}$$

Position signals are used because more reliable with respect to the others.

3.2.2 Result Analysis

Table 1 Comparison of speed (in frame per seconds) and reliability (percentage of frame identified with respect to total—21 599)

	ARTool	ARUCO
Speed	114.5 fps	94.3 fps
Reliability	98.9 % (21 380)	86.8 % (18 739)

The benchmark trajectory in space and its projection along the principal direction is depicted in Fig. 7. The reference frames of the markers are also presented.

ARUCO localization presents instability, and in different occasions it is not able to reconstruct the pose of the markers. In particular, between the frame 5672 and 5807 it completely loses the tracking—i.e. the spikes in figure. For further analysis,

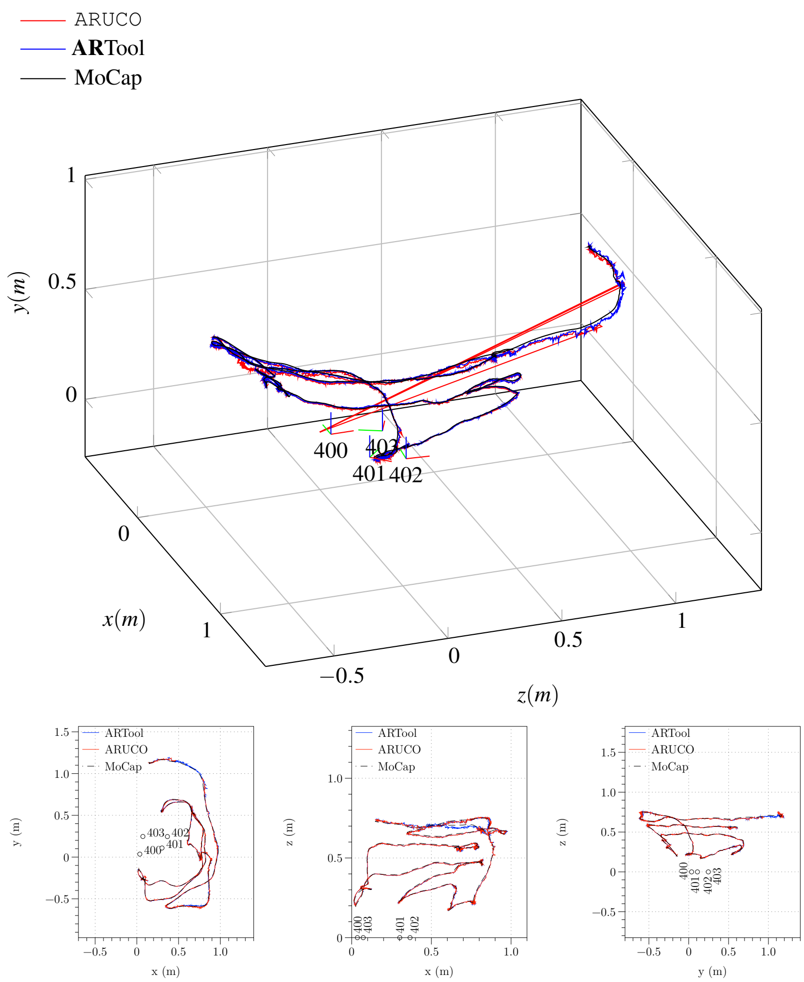


Fig. 7 On top of the image the 3D representation of the trajectories in the video. The reference frames of each marker are also reported. The spikes in the `ARUCO` trajectory are due to missed identification of markers

the `ARUCO` missing trajectory is approximated linearly between the last known and the first new localization. However this segment is the main cause of the differences reported in Tab. 1, where it is noticeable the reliability of `ARSceneDetector` ,

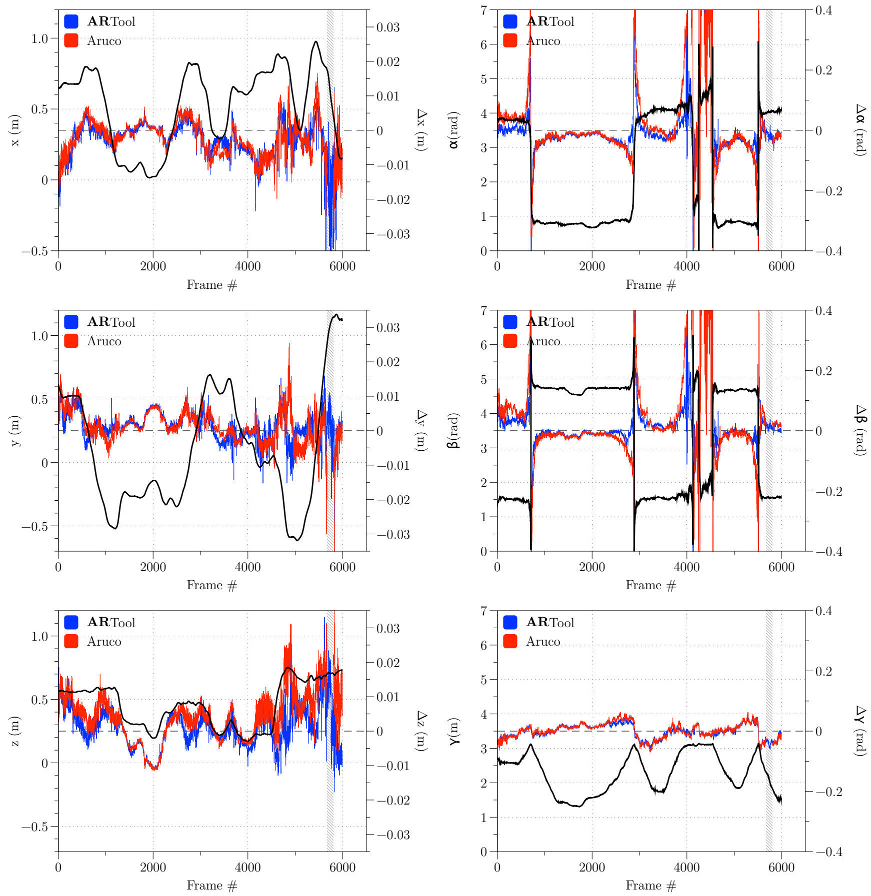


Fig. 8 Ego-localization errors. On the left, there are position plot and errors between marker libraries and MoCAP. On the right Euler's angles and their errors are plotted. `ARUCO` fails the identification between frames 5672 and 5807 (vertical hatched band)

that almost never drops track of the marker, scoring a quite high reliability index (98.9 %).

Fig. 8 shows a comparison of the three trajectory and the error of the markers detected trajectories with respect to the MoCAP one. In Fig. 9, histograms report the probability distribution for errors. For what concern positions, the error distribution

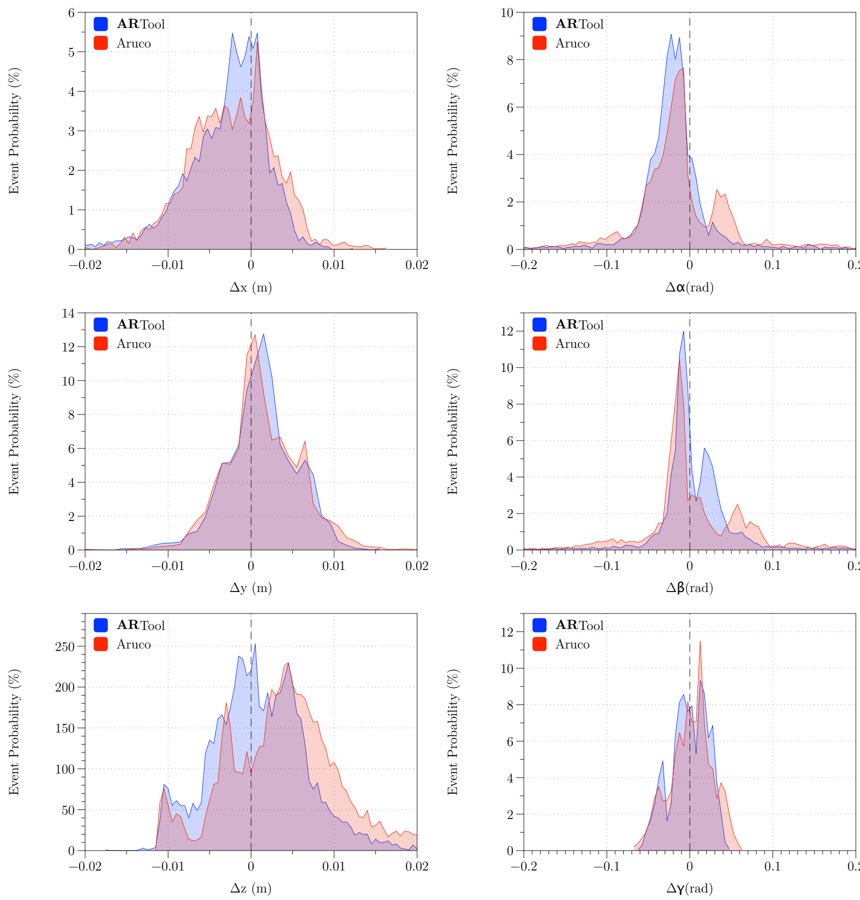


Fig. 9 Ego-localization errors distribution. The left column contains positions, while the right column contains Euler's angle

of ARUCO tends to be larger with a mode that diverges slightly from zero. Numerical analysis is reported in Tab. 2. Regarding attitude estimation, the performance can be considered comparable.

Table 2 Statistical indicators for errors distribution (mean μ, standard deviation σ and kurtosis k)

	Artool			ARUCO		
	μ	σ	k	μ	σ	k
x (mm)	-3.10	5.38	9.72	-6.04	2.93×10^{1}	7.58×10^{1}
y (mm)	1.22	4.33	8.75	4.05	2.05×10^{1}	5.85×10^{1}
z (mm)	9.37×10^{-1}	5.66	3.59	4.72	8.20	5.67
α (rad)	1.92×10^{-2}	2.99×10^{-1}	1.33×10^{2}	4.07×10^{-2}	3.09×10^{-1}	1.09×10^{2}
β (rad)	-9.07×10^{-4}	2.23×10^{-2}	2.38	-4.20×10^{-3}	5.17×10^{-2}	4.75×10^{1}
γ (rad)	2.13×10^{-2}	3.84×10^{-1}	1.52×10^{2}	1.67×10^{-2}	3.39×10^{-1}	1.53×10^{2}

4 Applications Showcase

The section presents a series of applications designed to test the most prominent capabilities and features of the **AR**Tool framework, leaving aside the authoring tools for machine manufacturer and technical offices.

4.1 Origin Debugger

The application visualizes the origins and the coordinate systems of both markers and numerical controller. The machine client is connected to the Heidenhain iTNC 530 of a Deckel Mori DMU-60T (5-axis milling machine), and takes advantages of an FTP connection for data exchange. From the FTP, the machine client downloads the iTNC file that stores the reference table. The file is queried at constant interval and parsed only if modification time changes.

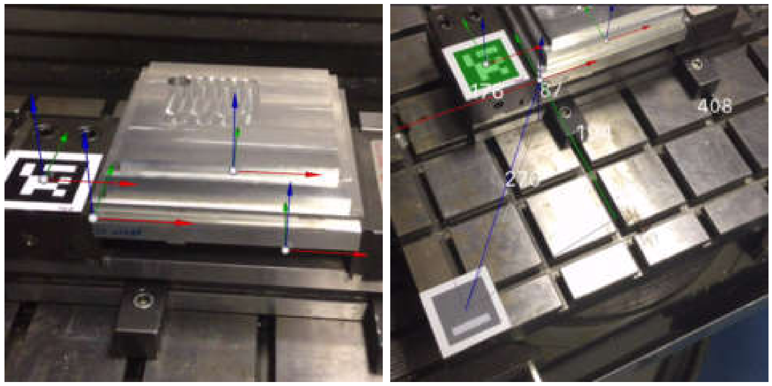


Fig. 10 The Origin Debugger application: on the left, the visualization of the reference frame obtained from the machine client, while on the right a measurement between different origin is performed

The application permits to see selected origins of the table projected on the screen, overlaid on the frame captured by the camera. Operators can inspect the scene from different orientations. The applications shows also distances between origins for debugging (see Fig. 10).

4.2 Trajectory Inspector

The application is built upon the capabilities of the previous application. The machine client queries the controller for the currently active part-program and download it through the FTP connection, alongside origin and tool table.

The tool table is parsed, and the name is used as identifier for the digital model to render, distributed through SCADA server. Since there is no communication channel for the numerical interpolator of this particular machine, it is the fallback

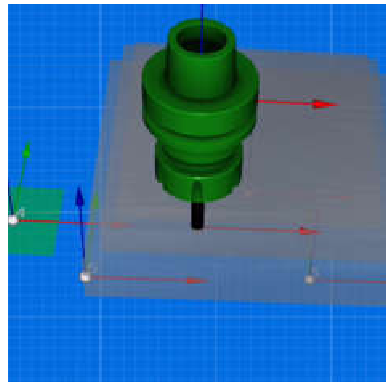


Fig. 11 The Trajectory Inspector: operator can navigate the virtual enviroment or fix it through a marker

ARTool interpolator that parses the part-program source file and generates the tool trajectory for the simulations.

Simulations are projected in a virtual environment that can be navigated by a user, exactly like a common CAE environment. It is also possible to fix the virtual environment through a marker and explore the simulation by moving and reorienting the tablet, as depicted in the screenshot of Fig. 11.

4.3 Trajectory Simulator

This application acts exactly like the Trajectory Inspector, and uses machine client and SCADA server to collect data and generate a virtually simulated environment that, in this case, is projected upon the camera feed. Operators can inspect directly the simulation in the working area, against real objects, the result of the interpolated trajectory and intercept collisions, programming errors, and misalignments (see Fig. 12).

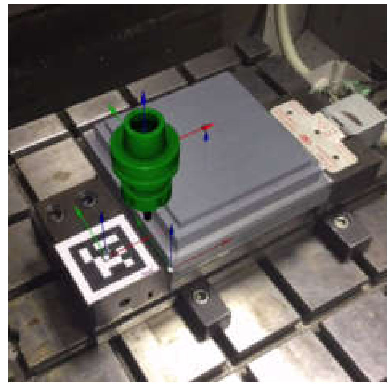


Fig. 12 The Trajectory Simulator: operator can navigate the virtual environment or fix it through a marker

4.4 ARTool Zero (Concept)

ARTool Zero is the concept of an Augmented Reality application that allows operator to select directly some geometric features as reference through the touching probe of a machine tool. Leveraging the input capabilities described in Section 2.4, the approximated feature information input through the augmented interface is transformed on-the-fly in a part-program that allows the touching probe to precisely identify the geometry.

Before performing the actual machine movements, a simulation of the trajectory of the touching probe is presented on the device screen, in Trajectory Simulation mode, so that operator can check for collisions, with respect to different point-of-view. The part-program generated is then loaded in the machine tool controller for the actual execution. A sequence that exemplify the usage is depicted in Fig. 13.

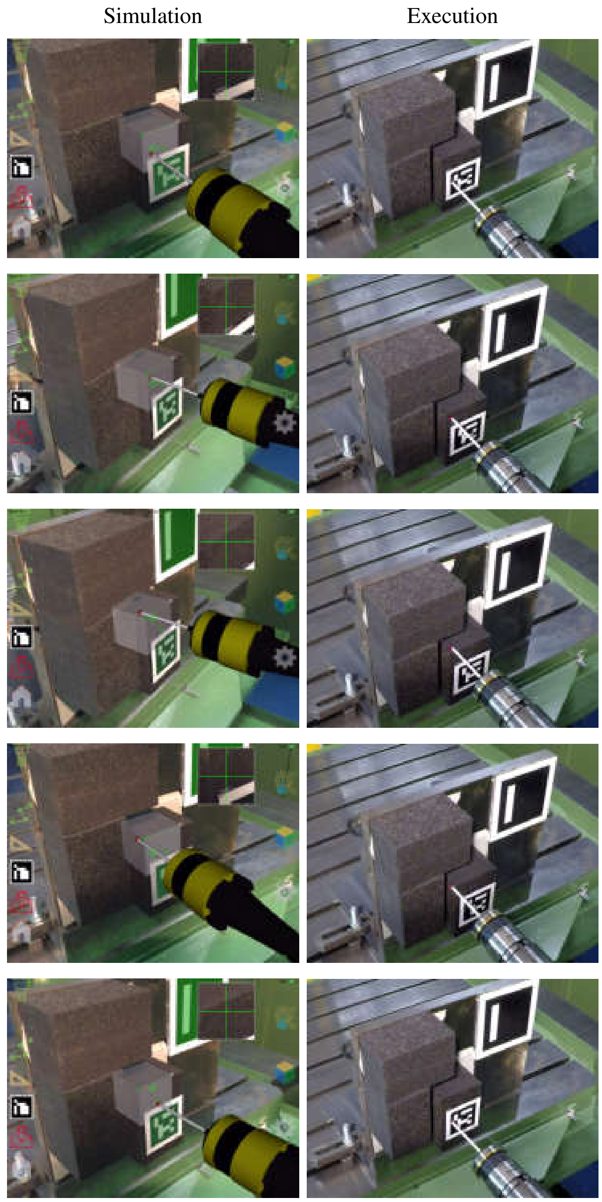


Fig. 13 Artool Zero Concept: the sequence on the left shows the simulated part program on the display of the tablet: users can frame the scene from different directions to check for collisions; on the right, the actually sequence of operations are depicted

Fig. 14 Maintenance Mode: a failed component highlighted

4.5 Maintenance Mode

The maintenance mode is at an early developing stage. The application requires a series of marker installed in the different parts of the machine to allow contextualized information gathering. In this case, the placement of assets on the screen does not require the same accuracy as in simulation, and a single marker covers quite a big area of the machine.

If a component fails the diagnostic, it is highlighted (see Fig. 14) and it is made evident to the operators. At the same time, an operator recall the manual page of a particular component by framing and taping it on the screen (using the input capabilities described in Section 2.4).

5 Conclusions

The work presents an Augmented Reality software framework for supporting CNC machine tool operations, such as setting up and checking for errors in part-programs or remotely guided maintenance operations. The systems uses a portable device (an Apple iPad) that overlays information to camera images by the mean of solid models and localized text. The system can work according to two prominent scenarios: *setup-mode* and *maintenance-mode*.

In *setup-mode*, the system shows workpiece shape and position, part-program simulated trajectory, and CNC setup data (reference systems, toolpaths, etc.). The mismatch between 3-D scene and real image are easily perceived by the user, that can *quickly* and *reliably* identify (and then correct) misalignments, collisions, and other errors in part-programs. But the framework does not act only as output interface. Indeed, leveraging the communication with the machine, the mobile device can act as measuring instrument, that can identify workspace coordinates—for example, as shown in § 2.4, **AR**Tool can be used to define the *workpiece origin* by the mean of a *free marker*.

The framework is also used for another scenario, namely *maintenance-mode*, that feeds the operator with service information from machine manufacturer. Visualized data include position of failing components and service operation sequences—e.g. the manufacturer may request the operator to check the axes lubricant reservoir: instructing the portable device to draw a red 3-D model of the tank, localized in space and overlaid on the real object, the operator can quickly locate it without checking machine schematics.

The framework is characterized by three layers, developed for testing purposes. The augmented interface layer comprises iOS applications, that uses camera and inertial sensors to perform ego-localization of the mobile device. In particular, camera images are processed by the custom made library **AR**Tool, designed for high performance and high reliability in manufacturing environment, tested against state of the art competitor `ARUCO`. **AR**Tool proved faster and more reliable when comparing the two libraries against a motion capture ground-truth. The device communicates with the machine client through a server that queries system status, positions, reference systems and part-program to be presented on the augmented application. The server also acts as information exchange systems (SCADA server). On the upper layers there are technical offices, that provide part-program to be executed and models, and machine manufacturers, that provide augmented documentations and operation sequences. Information are authored through plugin for CAE software.

References

1. Apple Inc.: SceneKit Framework. `https://developer.apple.com/scenekit/` (2016)
2. Bleser, G., Stricker, D.: Advanced tracking through efficient image processing and visual–inertial sensor fusion. Computers & Graphics **33**(1), 59–72 (2009)
3. Bondrea, I., Petruse, R.: Augmented reality - an improvement for computer integrated manufacturing. Advanced Materials Research **628**, 330–336 (2013). DOI 10.4028/www.scientific.net/AMR.628.330
4. Bradski, G., et al.: The opencv library. Doctor Dobbs Journal **25**(11), 120–126 (2000)
5. Büttner, S., Sand, O., Röcker, C.: Extending the design space in industrial manufacturing through mobile projection. In: MobileHCI 2015 - Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, pp. 1130–1133 (2015). DOI 10.1145/2786567.2794342

6. Chong, J., Ong, S., Nee, A.c., Youcef-Youmi, K.b.: Robot programming using augmented reality: An interactive method for planning collision-free paths. Robotics and Computer-Integrated Manufacturing **25**(3), 689–701 (2009). DOI 10.1016/j.rcim.2008.05.002
7. Ćuković, S., Devedžić, G., Pankratz, F., Baizid, K., Ghionea, I., Kostić, A.: Augmented reality simulation of cam spatial tool paths in prismatic milling sequences. IFIP Advances in Information and Communication Technology **467**, 516–525 (2015). DOI 10.1007/978-3-319-33111-9_47
8. Doshi, A., Smith, R., Thomas, B., Bouras, C.: Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing. International Journal of Advanced Manufacturing Technology pp. 1–15 (2016). DOI 10.1007/s00170-016-9164-5
9. Elia, V., Gnoni, M., Lanzilotto, A.: Evaluating the application of augmented reality devices in manufacturing from a process point of view: An ahp based model. Expert Systems with Applications **63**, 187–197 (2016). DOI 10.1016/j.eswa.2016.07.006
10. Fang, H., Ong, S., Nee, A.: Interactive robot trajectory planning and simulation using augmented reality. Robotics and Computer-Integrated Manufacturing **28**(2), 227–237 (2012). DOI 10.1016/j.rcim.2011.09.003
11. Fang, H., Ong, S., Nee, A.: Robot path and end-effector orientation planning using augmented reality. In: Procedia CIRP, vol. 3, pp. 191–196 (2012). DOI 10.1016/j.procir.2012.07.034
12. Fiorentino, M., Uva, A., Gattullo, M., Debernardis, S., Monno, G.: Augmented reality on large screen for interactive maintenance instructions. Computers in Industry **65**(2), 270–278 (2014). DOI 10.1016/j.compind.2013.11.004
13. Gilbert, W.: Economics of machining. Machining theory and practice pp. 465–485 (1950)
14. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
15. Hou, L., Wang, X., Bernold, L., Love, P.E.: Using animated augmented reality to cognitively guide assembly. Journal of Computing in Civil Engineering **27**(5), 439–451 (2013)

16. Jozef, N.M., Miroslav, J., Ludmila, N.M.: Augmented reality aided control of industrial robots. Advanced Materials Research **1025-1026**, 1145–1149 (2014). DOI 10.4028/www.scientific.net/AMR.1025-1026.1145
17. Kalpakjian, S., Schmid, S.R., Kok, C.W.: Manufacturing processes for engineering materials. Pearson-Prentice Hall (2008)
18. Lowe, D.G.: Fitting parameterized three-dimensional models to images. IEEE Transactions on Pattern Analysis & Machine Intelligence (5), 441–450 (1991)
19. Martínez, H., Laukkanen, S., Mattila, J.: A new hybrid approach for augmented reality maintenance in scientific facilities. International Journal of Advanced Robotic Systems, Manuel Ferre, Jouni Mattila, Bruno Siciliano, Pierre Bonnal (Ed.), ISBN **1729**, 8806 (2013)
20. Meden, B., Knodel, S., Bourgeois, S.: Markerless augmented reality solution for industrial manufacturing. In: ISMAR 2014 - IEEE International Symposium on Mixed and Augmented Reality - Science and Technology 2014, Proceedings, pp. 359–360 (2014). DOI 10.1109/ISMAR.2014.6948488
21. Monroy Reyes, A., Vergara Villegas, O., Miranda Bojrquez, E., Cruz Snchez, V., Nandayapa, M.: A mobile augmented reality system to support machinery operations in scholar environments. Computer Applications in Engineering Education **24**(6), 967–981 (2016). DOI 10.1002/cae.21772
22. Nee, A., Ong, S.: Virtual and augmented reality applications in manufacturing. In: IFAC Proceedings Volumes (IFAC-PapersOnline), pp. 15–26 (2013). DOI 10.3182/20130619-3-RU-3018.00637
23. Olwal, A., Gustafsson, J., Lindfors, C.: Spatial augmented reality on industrial cnc-machines. In: Proc. SPIE, vol. 6804, pp. 680,409–680,409–9 (2008). DOI 10.1117/12.760960
24. Ong, S., Pang, Y., Nee, A.b.: Augmented reality aided assembly design and planning. CIRP Annals - Manufacturing Technology **56**(1), 49–52 (2007). DOI 10.1016/j.cirp.2007.05.014
25. Ong, S., Yuan, M., Nee, A.: Augmented reality applications in manufacturing: A survey. International Journal of Production Research **46**(10), 2707–2742 (2008). DOI 10.1080/00207540601064773

26. PTC Inc.: Product Lifecycle Management (PLM) Software. `http://www.ptc.com/product-lifecycle-management` (2016). Accessed on June 26, 2017
27. Ramrez, H.b., Mendoza, E., Mendoza, M., Gonzlez, E.: Application of augmented reality in statistical process control, to increment the productivity in manufacture. In: Procedia Computer Science, vol. 75, pp. 213–220 (2015). DOI 10.1016/j.procs.2015.12.240
28. Schaumlöffel, P., Talha, M., Gorecky, D., Meixner, G.: Augmented reality applications for future manufacturing. Proceedings of the 5th Manufacturing Science and Education-MSE **1**(5), 2–5 (2011)
29. Schweighofer, G., Pinz, A.: Robust pose estimation from a planar target. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(12), 2024–2030 (2006)
30. Setti, A., Bosetti, P., Ragni, M.: ARTool - Augmented Reality Platform forMachining Setup and Maintenance, vol. 1, pp. 273–281. Springer (2016)
31. Surez-Warden, F., Mendvil, E., Ramrez, H., Garza Njera, L., Pantoja, G.: Mill setup manual aided by augmented reality. In: Mechanisms and Machine Science, vol. 25, pp. 433–441 (2015). DOI 10.1007/978-3-319-09858-6_41
32. Syberfeldt, A., Danielsson, O., Holm, M., Wang, L.b.: Dynamic operator instructions based on augmented reality and rule-based expert systems. In: Procedia CIRP, vol. 41, pp. 346–351 (2016). DOI 10.1016/j.procir.2015.12.113
33. Systems, D.: Solidworks Model Based Definitions. `http://www.solidworks.it/sw/products/technical-communication/packages.htm` (2016). Accessed on June 26, 2017
34. Vignais, N., Miezal, M., Bleser, G., Mura, K., Gorecky, D., Marin, F.: Innovative system for real-time ergonomic feedback in industrial manufacturing. Applied Ergonomics **44**(4), 566–574 (2013). DOI 10.1016/j.apergo.2012.11.008
35. Wang, X., Love, P.E., Kim, M.J., Park, C.S., Sing, C.P., Hou, L.: A conceptual framework for integrating building information modeling with augmented reality. Automation in Construction **34**, 37–44 (2013)

36. Wang, Z., Ong, S., Nee, A.: Augmented reality aided interactive manual assembly design. The International Journal of Advanced Manufacturing Technology **69**(5-8), 1311–1321 (2013)

37. Weinert, K., Zabel, A., Ungemach, E., Odendahl, S.: Improved nc path validation and manipulation with augmented reality methods. Production Engineering **2**(4), 371–376 (2008). DOI 10.1007/s11740-008-0115-3

38. Wójcicki, T.: Supporting the diagnostics and the maintenance of technical devices with augmented reality. Diagnostyka **15**(1), 43–47 (2014)

39. Zhang, J., Ong, S., Nee, A.: A multi-regional computation scheme in an ar-assisted in situ cnc simulation environment. CAD Computer Aided Design **42**(12), 1167–1177 (2010). DOI 10.1016/j.cad.2010.06.007