Retargeting Video Tutorials Showing Tools With Surface Contact to Augmented Reality

Peter Mohr[1], David Mandl[1], Markus Tatzgern[2], Eduardo Veas[3], Dieter Schmalstieg[1] and Denis Kalkofen[1]
[1]Graz University of Technology [2]Salzburg University of Applied Sciences [3]Know Center GmbH

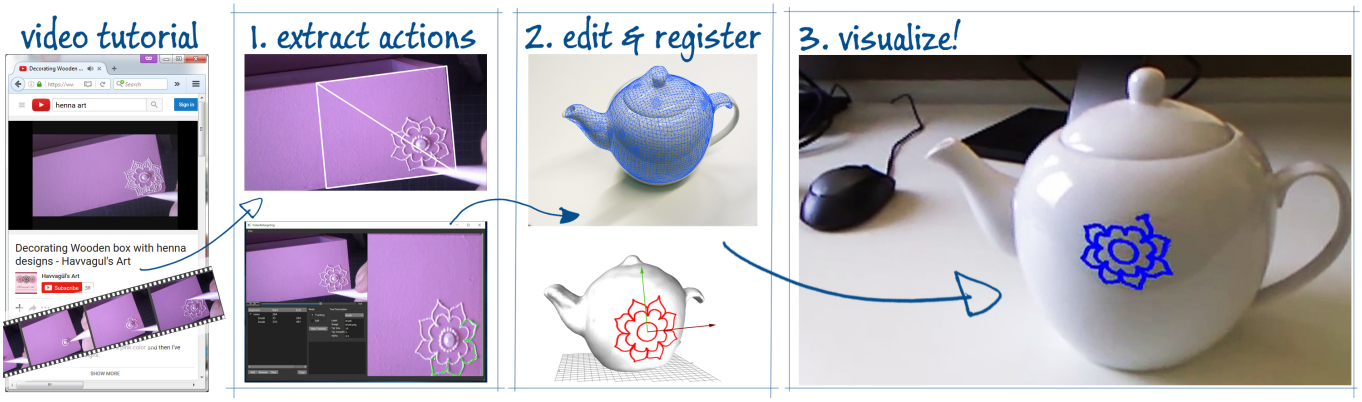

Figure 1. Retargeting a 'henna decoration' video tutorial to a teapot decoration scenario in the user's workspace. (left) The user extracts relevant motion from the video, (middle) scales it and aligns the result to a 3D scan of a teapot in the current workspace. With our editing tools, the user can quickly alter the original tutorial to meet their requirements. In this example, the original video tutorial shows a decoration consisting of dots, which requires a special henna pen. The user chooses to connect the dots into lines which can be drawn with a ceramic pen on the teapot. The user also scales down the entire ornament to better fit the desired aesthetics. (right) Using augmented reality, the user validates the result directly on the real teapot.

ABSTRACT
A video tutorial effectively conveys complex motions, but may be hard to follow precisely because of its restriction to a predetermined viewpoint. Augmented reality (AR) tutorials have been demonstrated to be more effective. We bring the advantages of both together by interactively retargeting conventional, two-dimensional videos into three-dimensional AR tutorials. Unlike previous work, we do not simply overlay video, but synthesize 3D-registered motion from the video. Since the information in the resulting AR tutorial is registered to 3D objects, the user can freely change the viewpoint without degrading the experience. This approach applies to many styles of video tutorials. In this work, we concentrate on a class of tutorials which alter the surface of an object.

ACM Classification Keywords
H.5.1 Information Interfaces and Presentation: Multimedia Information Systems - Artificial, augmented and virtual realities

Author Keywords
Augmented reality;virtual reality;retargeting;video tutorial

CHI 2017, May 06 – 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05...$15.00.
DOI: http://dx.doi.org/10.1145/3025453.3025688

INTRODUCTION
With the success of video sharing platforms, the production and distribution of homemade video tutorials is rapidly increasing, resulting in a large body of video tutorials available for nearly every aspect of life. Videos allow the demonstration of complex actions required to solve a certain task. Video tutorials are powerful in communicating motions quickly, but precisely following the instructions can be very challenging: Users have to match objects in the video with corresponding objects in their real environment. They must infer 3D motion paths, speed and velocity only from 2D video cues. In addition, object appearances in the video may differ from the real world, making it difficult to identify matching landmarks. The problem is exacerbated by the fact that the user's viewpoint often deviates from the one in the video.

The separation of task locations between the video screen and the real environment requires mentally complex hand-eye coordination [6]. To overcome this coordination problem, augmented reality (AR) tutorials present visual instructions directly registered in the user's real environment [38]. It has been shown that AR can significantly reduce the cognitive load required to follow instructions [20].

Unfortunately, *authoring* AR tutorials is a time-consuming process, which requires skills in 3D modeling and animation, in addition to mastering the technical components of an AR system. Consequently, few AR tutorials – certainly much fewer than video tutorials – exist today. In this paper, we ad-

dress this shortage with a novel system capable of retargeting homemade video tutorials to AR. We concentrate on a class of tutorials showing tools operating on object surfaces. This kind of tutorial describes actions that alter the surface of an object. Common examples are: painting, calligraphy, soldering or isolating circuits, make-up, and decorating, e.g., a teapot as shown in Figure 1.

Unlike previous work [16, 39], we do not simply overlay the video on the real object. Overlaid videos are efficient to produce, but not as effective as 3D tutorials, if the action is view-dependent or if the object in the video slightly differs from the one available to the user. In addition, a video overlay may clutter and occlude the real object, especially in surface manipulation tutorials, and animations in the video may distract the user while following the tutorial [44].

Therefore, our system extracts 3D motions from the video and registers the results to the 3D objects in the user's environment. This enables the user to freely change the viewpoint without degrading the quality of the AR experience. Moreover, this allows the presentation of instructions using effective illustrative visualizations, such as dynamic glyphs [36]. Illustrative visualizations are able to convey important information in an effective way with minimal clutter and, if necessary, without showing an animation. They also enable quick previews of arbitrary aggregate actions.

Furthermore, extracting 3D motions allows the editing of the tutorial's temporal structure. This may involve changing or adding actions or mixing multiple video tutorials into a novel one. For example, the tutorial in Figure 1 demonstrates drawing thick dots. However, a very specific tool is required to create thick dots, which the user might not have. Therefore, our system allows the connection of these dots into continuous lines representing the overall shape of the decoration. The result can be reproduced with a conventional flat drawing pen commonly used to decorate ceramics.

Our work provides three contributions.

- We provide the first system for retargeting videos showing tools with surface contact to AR. It includes a novel approach to capture motion with 3D surface contact from tracked 2D trajectories using registered 3D models, and a set of visualization techniques for conveying instructions in AR.
- We have conducted a user experiment of our system, from which we derive information about acceptance and performance, including an iteratively designed visualization for AR instructions.
- We discuss the components of our system, and we identify requirements and improvements for the design of similar systems.

RELATED WORK

Authoring computer-based tutorials traditionally involves the creation of dynamic glyphs, i. e., graphical elements, to present the path and direction of motions [36]. However, manual creation is very time and cost intensive, research has aimed at automating the authoring process. Automatic generation of AR instructions goes back to the pioneering work on KARMA [15], which is based on the idea of using rules to derive graphical representations, as proposed by Seligman and Feiner [41]. However, KARMA relies on a manually created knowledge database to derive its visualizations.

Script languages, which capture visualizations in procedural form, have been proposed by multiple authors, such as Butz [8] and Ledermann [28]. However, authoring by scripting always requires formal knowledge about the tutorial content in addition to programming skills. Therefore, systems which require less user input have drawn attention. Agrawala et al. [1], Li et al. [30], Kalkofen et al. [22] and Kerbl et al. [24] showed the automatic generation of disassembly instructions for rigid objects. Mohr et al. [33] demonstrated the automatic generation of 3D animations from images depicting an assembly sequence. All these systems derive a sequence of actions consisting of straight motions only. Therefore, they are unsuitable for tutorials which involve complex motions.

The authoring of more complex actions has been proposed by capturing the necessary steps. For example, Grabler et. al demonstrate the generation of photo manipulation tutorials [17] from recorded actions in a photo-manipulation tool, and Chi et. al mixed 2D image and video material to generate tutorials [9]. Our system follows the idea of recording and replaying actions and lifts it to 3D AR environments.

AR tutorials have already been generated using 3D motion capture systems. Examples include assembling furniture [46] and Lego toy-sets [18], or gestural commands [45]. While 3D motion capture is the obvious way of providing input, these systems handle rather simple motions. However, the very recent work from Chi et al. [10] captures more complex motions from which the system generates illustrative step-by-step diagrams. While the generated drawings are optimized for effective communication, the system presents the results in 2D, rather than registered in 3D AR. Similarly, AR instruction systems often make use of the idea of a 2D mirror. Examples include AR instructions for physiotherapy [43] and dance [2] applications. While this concept has been demonstrated to be effective, it is limited to body centric instructions.

Several works [16, 39, 27, 13] forgo the use of 3D capturing and, instead, overlay recorded 2D video directly in the AR display. Using 2D video has the advantage that the original demonstration is preserved without requiring any spatial or semantic interpretation. However, showing a registered 2D video does not allow a truly free choice of the viewpoint. Moreover, the video occupies substantial screen space. Both severely restrict the practical value. In contrast, the system of Damen et al. [13] provides the video tutorial using a heads-up display which allows clearly seeing all objects. However, no direct augmentation is given why the user has to mentally match landmarks between real objects and video data.

The methods in this paper are also inspired by research on extracting image layers [42] and motions from video data [25, 12]. However, while these systems aim to generate editable static 2D visualizations, our system allows editing entire sequences which are going to be presented within an interactive 3D environment.

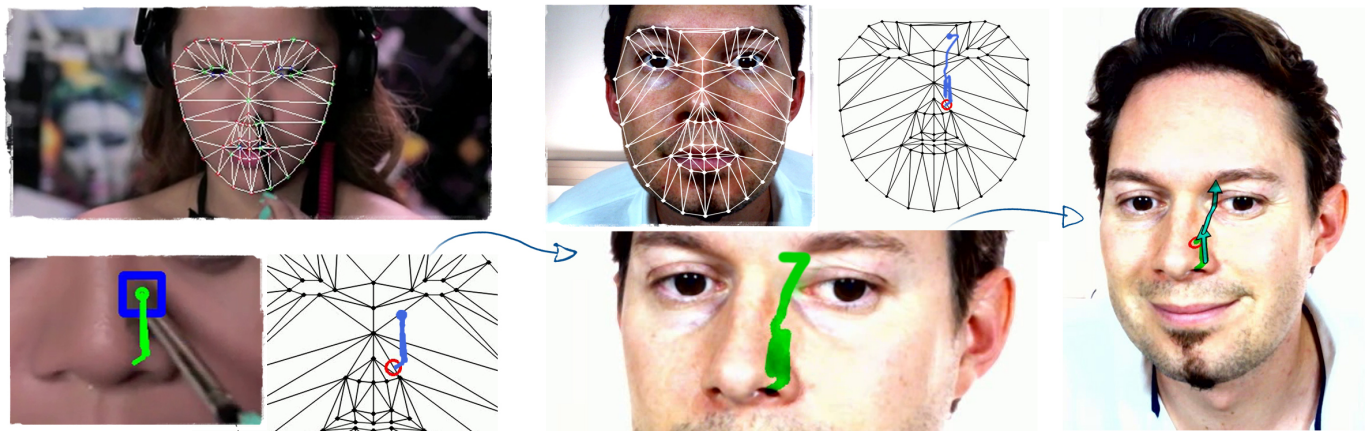

Figure 2. System overview. (left) We extract object and user motions by tracking known model features in the 2D video. Here, tracked features are used to record the path of the brush and to align a face model in each frame. (middle) After validating and possibly editing the extracted motion, we retarget the motion data to real world 3D objects. This requires registering the same 3D model as used in the extraction stage, in this case, a face model, to the live camera image. By tracking the model in 3D, we are able to relate video data to the real world. In this example, we present the recorded path of the brush directly on the user's face. (right) Since we retarget the extracted motion data in 3D, we can choose an arbitrary point of view. To provide effective visual instructions, we generate dynamic glyphs (here: timed arrows) and we indicate the position of the brush over time using a red circle.

OVERVIEW

We begin with an overview of our system. It is designed to let an author quickly extract the relevant information from a source video and compose an AR tutorial that operates in the environment of the user. All steps can be done by a single person, but there are two clear roles: the expert editor and the actual consumer of the tutorial. The extraction phase is more suited for an expert, i.e. the person who generates the video tutorial. The editing/retargeting phase is about adapting the tutorial to the object of choice. This is a preparation task that the end-user might do prior to consuming the instructions. If one has the same object no adaptation is necessary, otherwise the tutorial has to be aligned (i.e. retargeted) to the new object.

Step 1 – Extraction The user loads the input video. The system either recognizes the main object (such as the face in Figure 2), or the user interactively models the 3D object of interest. The user selects the tip of the tool from which the motion is tracked and extracted. If the system loses tracking, the user is asked to insert a cut or to reinitialize the tool tracker.

In Figure 2 (left) the system recognizes the face and automatically registers a deformable face mesh. Afterwards, the user marks the tip of the make-up brush in the first frame of an action. The system is now able to track the brush within each frame of the action. We map the tracked 2D trajectory of the brush onto the 3D mesh of the face, which retargets the 2D into a 3D trajectory. The mapping is implemented by automatically unwrapping the 3D mesh into 2D texture space. This allows us to directly add the 2D tracked trajectory of the brush to the 2D texture of the face.

Step 2 – Editing The extracted motions are validated, corrected where necessary, and registered to the user's real world object. By tracking the 3D model of the object in the video, the extracted motions are registered automatically. However, we allow the repositioning and reorientation of the extracted motion, and we provide tools for combining multiple sources into new tutorials.

In Figure 2 (middle), we retarget the trajectory of the make-up brush by registering the face mesh along with its previously generated 2D texture to the user's face.

Step 3 – Visualization The motion is presented using an effective visualization based on arrows to indicate direction and position. Optionally, our system is able to show an animation of the tip of the tool along the extracted 3D trajectory. However, based on user feedback we recommend to use animations only if mimicking the speed and velocity of the original motion is important.

In Figure 2 (right), we abstract the motion by using one arrow for each segment of the trajectory. Note that, based on user feedback we slightly refined the visualization shown in Figure 2 (right), resulting in a presentation which uses arrows along the outline of the trajectory (see Figure 7).

EXTRACTION

In order to extract motion with surface contact, we first extract the 3D motion of the surface by tracking the corresponding 3D object. Subsequently, we extract the motion of the tip of the tool relative to the surface.

Tracking the 3D motion of known objects from monocular videos is a standard task in AR, assuming that a 3D model of the object and intrinsic camera parameters are available. For objects with enough surface texture, a popular approach is to match SIFT features [31] extracted from the live video with a 3D feature point cloud representing the object. The 3D positions associated with matched features are forwarded to pose estimation using a variant of the Perspective-n-Points (PnP) algorithm, such as the one of Lepetit et al. [29]. After successfully computing the pose of an object in the first frame, pose estimation in subsequent frames can be made faster by incremental tracking [32].

Extracting the 3D motion from arbitrary videos from, e.g., online sources is more difficult, since the 3D model and the intrinsic camera parameters are unavailable. Therefore, we must

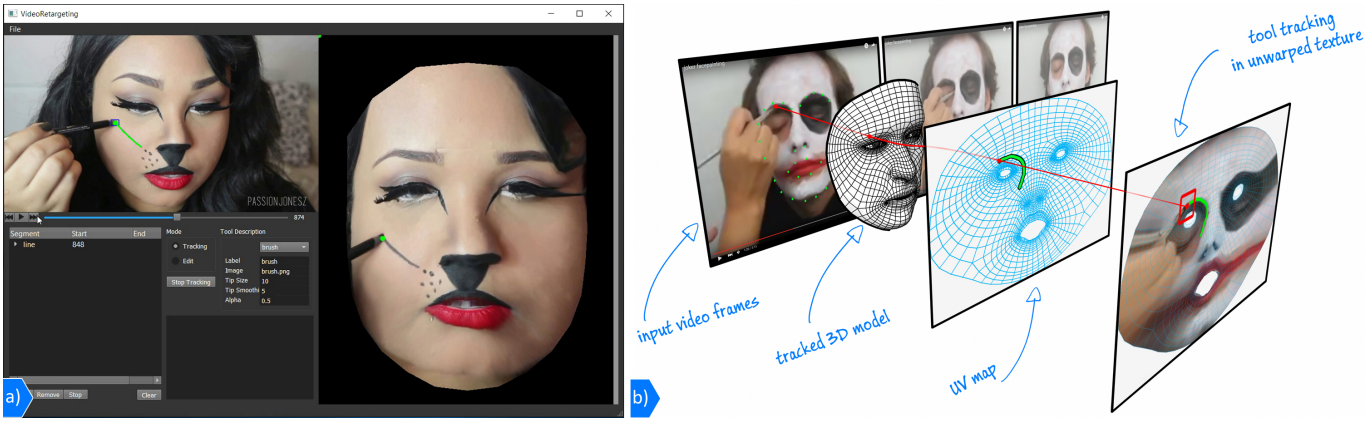

Figure 3. Extracting motion with surface contact. (a) We extract the 2D trajectory by tracking the tip of the tool in unwrapped texture space. (b) We convert the 2D trajectory to 3D by back-projecting the video data to a corresponding 3D model, in this case, a face.

interactively create a 3D model and adjust internal camera parameters as part of the authoring process.

Our system extracts 3D motion of three different types of 3D objects. In particular, we provide tools to extract 3D motion of piecewise planar objects, rigid objects, and deformable objects with a known shape model, such as human faces. For each of them, we provide an optimized set of tools to create the necessary 3D model with minimal user input.

Planar objects

Piecewise planar objects, such as planes or boxes, can be conveniently specified by interactively drawing on top of the first video frame. We let the user specify the corners of a rectangular area and its dimensions. Note that a minimum of four points is sufficient to estimate a pose from a homography. However, to produce more stable results, we incrementally track all distinctive features inside the rectangular area throughout the video sequence. The locations of the distinctive features in the plane are estimated directly during the extraction from the image.

Non-planar objects

Non-planar 3D objects require a more complex 3D point cloud than a tracking model. Unfortunately, for most online videos, we cannot expect to successfully perform structure from motion to obtain 3D geometry: The internal camera parameters are unavailable and may even change, when optical zoom is used. The objects in the video may lack texture features. Visibility may be limited due to restricted camera motion. Occlusions, such as from the author's hands, may be significant.

Instead, our method utilizes the fact that a user wishing to replicate a tutorial in AR must have access to the same class of objects as the ones used in the video. While actual shape and appearance may differ between the object in the video and the one available to the user, the overall topological and geometric characteristics will be similar. Therefore, we let the user provide a physically available object as a template for the tracking model.

Creating a 3D reconstruction of an existing physical object is relatively straight forward using consumer depth sensors [35] or even mobile phone cameras [37]. If no physical template is available, the user can instead search for a template model in an online database, such as Sketchup 3D Warehouse[1]. If the template model slightly differs from the tracking model, we incrementally deform based on the approach of Kraevoy et al. [26].

Deformable objects

If the shape is known a priori, such as the human face in Figure 2, a deformable model can be tracked by determining an update to the deformation parameters in each frame. This kind of tracking is commonly based on specially trained models. Our system implements facial model deformation based on a constrained local neural field [3]. Since the face tracker operates in image space, we map the 2D facial landmarks to a 3D model of a human face (see Figure 3).

Motion Capture

To extract the path of a tool which alters the workpiece (i. e., the 3D surface), we extract the trajectory of its tip. The user has to select starting and ending frames of an action in order to input information about surface contact. In-between the selected frames we assume the tool has contact to the surface. To extract its motion, we track the position of the tool relative to the workpiece, and we track the workpiece in 3D space as described before. This enables us to extract 2D trajectory in image space which we subsequently convert to a 3D trajectory by using perspective projection from the tracked camera's point in space onto the 3D surface (Figure 3). Note that we have derived the 3D pose of the object relative to the camera for every frame before. This allows the derivation of the camera pose as the inverse of the 3D object pose transformation.

Since the tool's tip is often very small and usually located in front of a changing background, tracking the tip from monocular video can be difficult without the exact knowledge of the tool's appearance. Therefore, conventional feature detection approaches do not yield satisfying results. We overcome this problem by applying a tracking-learning-detection approach [34], which identifies robust features by updating a

[1] www.warehouse.sketchup.com

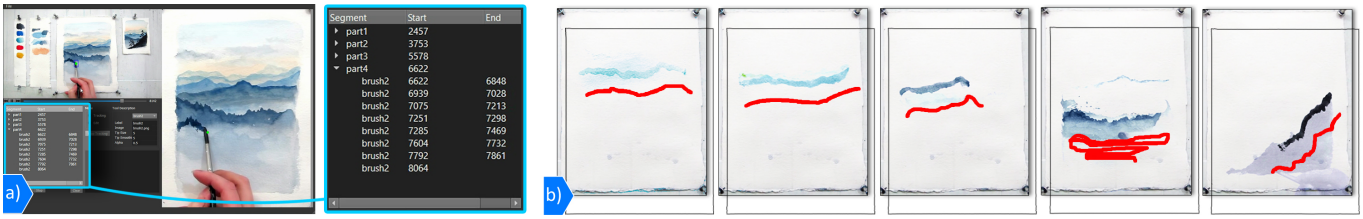

Figure 4. Segmentation and Layering. (a) We interactively segment the input data by selecting starting and ending frames. (b) This results in a set of actions (red), which we can use to derive image layers.

Table 1. Tool tracking accuracy measurements.

unit:mm	pencil	felt-tip	marker	brush
tip-size (lxw)	2x0.5	4x2	7x5	24x6
stroke-width	0.5	1.2	2.2	10-14
error (avg)	0.17	0.23	0.37	4.47

learned classifier, while tracking an initial user selection. In order to track a tool, the user identifies its tip by selecting a rectangular patch in the first video frame (marked red in Figure 3(b)). This area is used to initialize the classifier, which is improved in subsequent frames. As proposed by Kalal et al. [21], we use an ensemble classifier for detection and P-N learning to update the tracking model.

After extracting the path of the tool in image space, we project its position to the surface of the workpiece. We have implemented this by recording the tool trajectory in a 2D texture atlas, generated by unwrapping the 3D mesh of the workpiece. The texture coordinates allow us to directly map the tool trajectory from image space to the surface.

Accuracy Analysis

The accuracy of the tool tracker depends on the stroke width, tip size and speed. While the tracker follows the tip it might not follow the actual center of the stroke. The author can directly adjust this offset during/after extraction. Table 1 shows tracking accuracy measurements of a variety of drawing tools. The watercolor brush has the largest error due to its large and deformable tip, while all other tools produced an average error of less than 0.5 mm.

EDITING

After extracting objects and their motions, an editing pass lets the user arrange the content into the form necessary for AR presentation. This mainly concerns the temporal structure (motion segments) and the spatial arrangement (layers of affected surfaces).

We split the motions into temporal segments to allow convenient navigation, as proposed by Pongnumkul et al. [40]. For each segment, we compile the visual changes caused by the performed action into a corresponding image layer. The resulting set of layers allow us to edit the extracted motion, to realign them (i. e., reposition and reorient), and to create new compositions of multiple actions from possibly different tutorials.

Temporal segmentation

We define a temporal segment by an action in the input video, e. g., a brush stroke painted on a canvas. We carry out the segmentation semi-automatically, and we let the user refine the result at any point in time. More specifically, we ask the user to define starting and ending points of an action along with a meaningful name during extraction. We allow to group successive actions into a single level of a hierarchy, and we automatically refine the resulting segments based on an analysis of the motions of the tool within a segment. To refine segments, we identify turning points of the motion, i. e., points in the path were the angle between two neighboring line segments exceed a threshold. In our examples, we used an angle of 90° as threshold.

Layering

For each segment, we also extract a layer storing a video matte and the foreground colors which represent the changes between the starting and ending keyframes of a segment. We calculate the foreground color using the method of Chuang et al. [11] which intersects vectors between foreground and background colors in RGB space. Background colors are derived from pixels in future frames of the video which have been identified to be overwritten by the action.

After estimating background and foreground colors along the trajectory of an action, we calculate the alpha value as $\alpha = |B_1 - F|/|B_0 - F|$, where B_0 represents the background color in the starting frame of the segment, B_1 represents the background color in the ending frame, and F, the estimated foreground color. This method requires the presence of two different background colors B_0 and B_1. Therefore, if an insufficient color distribution in the background is present, we resort to simple chroma keying to estimate the foreground color. While this part of our system is similar to the work of Tan et al. [42] it extends it by allowing for a moving camera and it simplifies deriving layers based on tool interaction.

By using the segmentation and the mattes, we can generate a layered representation of the source video. A few selected layers of a painting tutorial are shown in Figure 4(c).

The layered representation can be manipulated in a way that is similar to common image editing software. Specifically, layers (and the embedded paths) can be re-arranged, scaled and combined. Multiple layers from different video tutorials can be mixed, if the depicted models are geometrically compatible. We support the Adobe Photoshop format for editing of layers in third-party tools.

Furthermore, our system allows any combination of layers to be rearranged on the real object. An initial registration of a layer is available from the extraction pass. The user can interactively reposition and reorient layers using a texture atlas corresponding to the surface of the real object. For example, the decorations shown in Figure 2 were originally applied to a jewelry box, but later retargeted to a teapot.

VISUALIZATION

To visually communicate instructions we generate graphical elements based on the extracted 3D motion. Following the work of Nienhaus et al. [36] the goal of our design was to introduce minimal visual clutter by providing abstractions of the motion. Therefore, in our initial design we create dynamic glyphs based on arrows which we combine with an animation of the motion. We present the animation using a red circle which marks tip of the captured tool. However, we iteratively optimized our design as users were not totally satisfied with its usability. We present the design iterations in the evaluation sections.

In all our visualizations we reduce the complexity of motion paths, as raw motion trajectories are often too cluttered and jittery to be suitable for path visualization (Figure 2, middle). Our filter simplifies a path by first using the approach of Douglas et al. [14](Figure 6(b)), followed by an additional segmentation of the paths into smaller segments. To segment the path, we search for turning points by comparing the angle between two neighboring line segments to a threshold (the example in Figure 6(c) uses a threshold of 90°). Subsequently, we cluster turning points which are placed close to each other by recursive merging based on distance. The resulting paths can be used to abstract the motion using arrows. We create an arrow head at each turning point and the endpoint (Figure 6(c)).

EVALUATING THE AUTHORING

We have tested our system on a number of different video tutorials. Throughout this paper, we present snapshots to discuss the tutorials. The video material supplied with this paper shows the results in greater detail.

We collected feedback on the authoring step for samples from a facial make-up and a painting scenario in an expert evaluation. Facial make-up tutorials include motions with surface contact on a 3D human face model. Figure 5 illustrates the video tutorial. The retargeting system automatically finds the face of the tutor. The author has to initialize the tool tracker in the first frame of every segment of the video and stop tracking in the last frame of a segment. In a few cases, the tracker had to be re-initialized after tracking failures.

The painting tutorial includes motion with surface contact on a canvas. Figure 4 illustrates the painting tutorial. We model the canvas as a planar object, and further extract actions and layers. We use Adobe Photoshop as an interface for editing, where we load the extracted layers automatically. Photoshop provides operations like translation, rotation or scale, allowing to modify the input tutorials. We can add layers from multiple tutorials to combine several source tutorials into a new one.

Four expert users, experienced in using image and video editing software, participated in the evaluation. Before starting, the participants familiarized themselves with the content of the videos, the task in the tutorial and the authoring tool. To collect feedback on the tool tracking, we asked the experts to extract the motion in two ways: first, by redrawing the line manually on the surface texture representation that shows the final result of an instruction; second, by using the tool tracker to extract the path of the tools automatically.

To comfortably extract start and end frames of instructions we allowed the modification of the input video frame rate by a scale factor. Based on our experience, we set the scale factor to 0.5, resulting in a time-scaled video. While users appreciated scaling the speed of the video playback, they also asked for an interactive control of the scale factor.

For the facial make-up sample, participants required approximately *four minutes for 50 seconds* of time-scaled video. For the painting tutorial, participants required around *eight minutes for three minutes* of time-scaled video.

The tool tracker was met with positive responses. Aside from occasional tracking failures, participants were comfortable using this tool. When directly compared to redrawing the instructions manually, participants considered the tracking was faster for extracting the instruction. Two participants even stated explicitly that it is more accurate. With respect to the tracking, one participant remarked that our results are precise enough to rather rely on the extracted original drawing than on a line she is redrawing manually.

Participants generally suggested more advanced tools, such as shortcuts to fill areas in paintings. The current authoring software relies solely on extracting paths. However, future work will investigate the use of collections of tools that allow the efficient extraction for actions in areas. A fill instruction could be specified in one frame and automatically extended to instruction steps over multiple frames.

EVALUATING EFFICIENCY OF AR MAKE-UP TUTORIAL

A second experiment was conducted with the goal to corroborate the precision of activities performed when following the AR tutorials. The intention was to observe the reaction of non-technical users in performing a common task aided by AR tutorials, and to compare the results with those obtained when following conventional video instructions.

Design. We introduced a structure for making comparisons. The study had two conditions: video (V) and AR. The AR condition showed the instructions step by step. We chose a face-painting task based on a video downloaded from Youtube (Figure 5(a)). The tutorial involved two types of precision tasks, namely painting points and painting lines. It had the advantage that it could be divided into two symmetrical parts: left and right side of the face.

The study was organized as a repeated measures design with two independent variables: interface (V, AR) and task (left, right side of face). Task was treated as random variable for counterbalancing the design so that each participant uses a different configuration. The possible configurations were

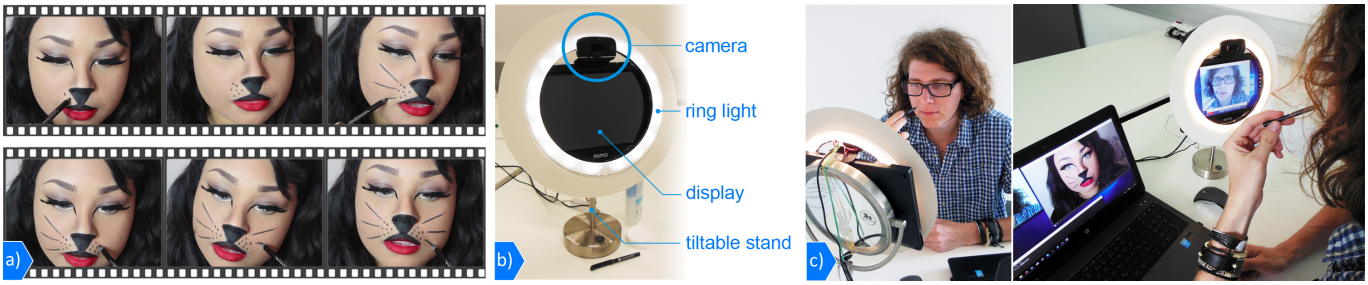

Figure 5. Experiment setup for a retargeted make-up tutorial. (a) Input video tutorial. (b) We showed the resulting AR tutorial using an AR mirror, which consisted of a camera and an USB display. (c) Participants could use the AR mirror and the video which we placed next to the mirror.

(AR_{left}, V_{right}), (AR_{right}, V_{left}), (V_{left}, AR_{right}), (V_{right}, AR_{left}). Participants were randomly assigned to configurations. A make-up AR-Mirror was built for the study, replacing a table-stand make-up mirror with a display (Mimo Magic-Touch, 10.1", 1024x600 pixels) and camera, shown in Figure 5(b). We control the AR visualization using a standard PC mouse and a next button, and we control the video using the interface of a common video player with functionality to scroll back and forth.

Pilot. We performed a pilot study with the described setup. Three female participants ($\overline{X} = 33$ years old) were asked to take part in the test. They signed a consent form accepting that their performance be video-recorded. Answers to a pre-test questionnaire indicated that one participant relies on make-up videos, whereas the other two had never followed video instructions for make-up before. The session was closed with the participants rating the difficulty of reaching for the controls and a semi-structured interview.

Having to reach for controls was not seen as hindrance either in AR ($M = 4.6$ of 7, higher means easier) or in V ($M = 4$). Participants commented on the lack of preview in AR ("There is no preview. I have no idea what I am trying to achieve, because I only see each individual instruction.") and on occlusion issues ("Occlusion. I cannot see my skin. The instruction is getting in the way, and I cannot see if I am painting it correctly or not.").

Revision and Experiment. After the pilot, the interface was modified as follows:

- Full preview was added for future steps of the tutorial.
- The visualization was modified to avoid occluding the user's skin. It displays only the outline of the motion trajectory and the motion is shown using an animated circle (Figure 6(d)). The arrow is still used to preview a segment's path, but it fades out after the preview phase (Figure 6(d)).

Six participants took part in the study (1 female, $\overline{X} = 34.3$ years old sd=4.8). The setup and the procedure were identical to the pilot. However, in contrast to the pilot study, we asked to follow the instructions as accurately as possible. Note that, while other researchers have compared monitor versus AR instructions [20], we were interested in the perceived quality of the instructions generated with our method. However, we also measured error and task completion time to evaluate the performance of our AR visualization system.

Results and Discussion. The task completion time was measured using a stop watch from the point in time where participants announced the start of the drawing. The drawing was considered finished when the last stroke was placed. The error was measured by normalizing the texture atlases containing the strokes of the tutorial make-up and the user-drawn make-up and comparing them using the l2-distance between the image pixels. Wilcoxon signed rank tests did not reveal any significant differences in time (V: mean=82.2s, sd=29, median=72; AR: mean=102.4s, sd=31.9, median=88) or error (V: mean=0.45, sd=0.04, median=0.47; AR: mean=0.42, sd=0.02, median=0.42).

We received overall positive feedback on the AR condition. Comments from the experiment included "I was more confident of being accurate when using AR.";"I felt I was quicker using AR. Being accurate with the video was difficult, because I didn't know where exactly I have to place the dots.";"The mirrored video was difficult to (mentally) invert.";"In AR I didn't have to think, I could concentrate on the drawing.".

All participants unanimously preferred AR over the video when the goal was being as accurate as possible. In a comparative questionnaire, they unanimously expressed feeling more confident and faster with the AR interface. In the video condition, two participants did not pick the correct side of the face, when starting the task. They were instructed to continue on the correct side. This was not an issue in AR, which showed instructions directly on the face of the participants. However, while the participants felt that AR allowed them to follow the motion exactly where they should appear, we noticed that lines drawn in the AR mode tend to include a bit more jitter. We believe that this is because in V people drew continuously while in AR they used a stop and go strategy which allowed them to repetitively validate the result. We see two possible reasons for this behavior. First, our 3D face tracker is not perfectly modelling face deformations. Therefore, the augmented lines were floating a bit over the skin of the user as soon as the face was deformed. This may have distracted the user resulting in a stop and go strategy. Second, our visualization encodes the speed of the motion using a moving dot. This may have distracted participants, because they switched their focus from being accurate in space to being accurate in time and the other way around, thereby causing them to continuously interrupt the motion.

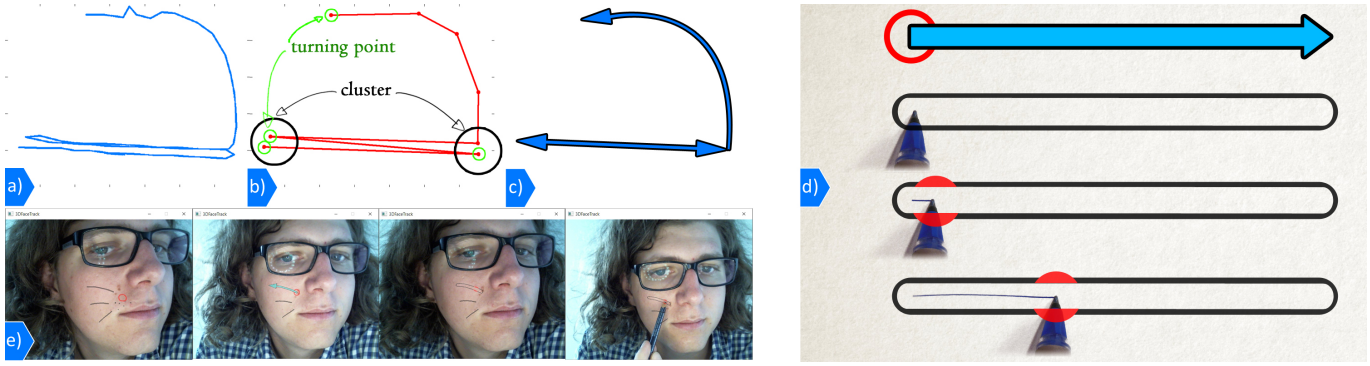

Figure 6. Path generation and first revision of AR visualization. (a) We generate path illustrations from motion capture data. (b) The extracted path data is analyzed and simplified. In particular, we remove zig-zag overdraw along the trajectory by clustering and detect turning points (marked in green). (c) We generate arrows in-between turning points, the start point and end point. (d) At runtime, we use the arrows to provide a preview of the motions. To minimize occlusion, the arrow is replaced by the border of the tool's trajectory. The red dot shows the extracted tool position over time. (e) The combination of visualization techniques provide an overview first, before the user can follow the exact motion.

Even though we received positive feedback on our AR interface, it did not outperform video based tutorials in task completion time or error rate. We believe we detected no differences due to the imperfect real world modelling of the user's face in AR and the distracting animation which was not necessary for the task.

EVALUATING EFFICIENCY OF AR KANJI TUTORIAL

After the make-up study, animations were removed from the interface. Instead, the direction of the motion is encoded in the border of the instruction glyph (Figure 7(c)). We performed a third experiment to collect quantitative data on the performance of the second revision of our AR visualization system. Since we speculate that the face tracking solution was not accurate enough to allow objective comparison of the quantitative data, we switched to a drawing scenario, in which accurate tracking and a rigid scene (without deformation) could be ensured. Therefore, participants were asked to follow Kanji drawing tutorials on paper using our AR interface and a common video interface.

In the AR condition, participants used an Optical See-Through Head Mounted Display (HMD), a Microsoft Hololens, to receive instructions augmented on a piece of white paper (Figure 7(a)). The Hololens has no standard mouse interface, therefore we had to change the AR interface so that switching to the next instruction was done using a handheld controller (Figure 7(a)).

Design. We designed a repeated measures within-subjects study to compare the performance and user experience of the AR interface to a common video interface. We introduced one independent variable interface with two conditions: AR interface (AR2) and video interface (V2). The task was to follow a tutorial with the goal to draw a single Kanji symbol in a target area as it was shown in the respective interface. The task was repeated ten times for each interface using a different Kanji symbol for each repetition. Correctly following the tutorial involves drawing strokes of the proper size, from the correct direction and in the right order.

We prepared a set of 20 symbols, which were of similar complexity (based on the number of strokes) consisting of 6 to 8 strokes. The 20 symbols were divided into two pools of ten symbols, each pool contained an equal total number of strokes. The tasks and the pools of symbols were counterbalanced to avoid learning effects and bias by the choice of symbols for each pool.

As dependent variables we measured task completion time and error of each task, subjective workload measured by the NASA TLX [19], usability on the System Usability Scale (SUS) [7] and overall preference.

Apparatus. Participants performed the task standing in front of a whiteboard and drawing with an ordinary pen on a $10cm \times 10cm$ piece of paper attached to the board (Figure 7).

In AR2, participants wore a Microsoft Hololens HMD and used the second revision of our visualization (Figure 7). In V2, an Nvidia Shield tablet ($17.2cm \times 10.8cm$) showing the instruction video was mounted above the target area (Figure 7(b)). Participants were presented the unmodified tutorial video and had to follow the instructions. The MX Player application[2] was used as video player. The unmodified tutorial videos did not show an overview at the beginning of the tutorial. Participants could browse through the video using common video controls (Figure 7(b)).

Procedure. After an introduction and filling out a demographic questionnaire, participants performed a training task for the first interface using a training symbol, different for AR2 and V2, that was not part of the tested set of symbols. After participants were comfortable using the interface, the measured tasks started and participants were instructed to be fast and accurate. The symbols from the current pool were shown in random order. After finishing the 10 symbols for one interface, participants filled out the NASA TLX and the SUS. The second condition started thereafter, following the same procedure. After filling the second SUS questionnaire, participants filled out a preference questionnaire and a semi-structured interview was conducted. A session took approximately 45m.

[2] goo.gl/xd5rb6, last accessed September 20^{th}, 2016

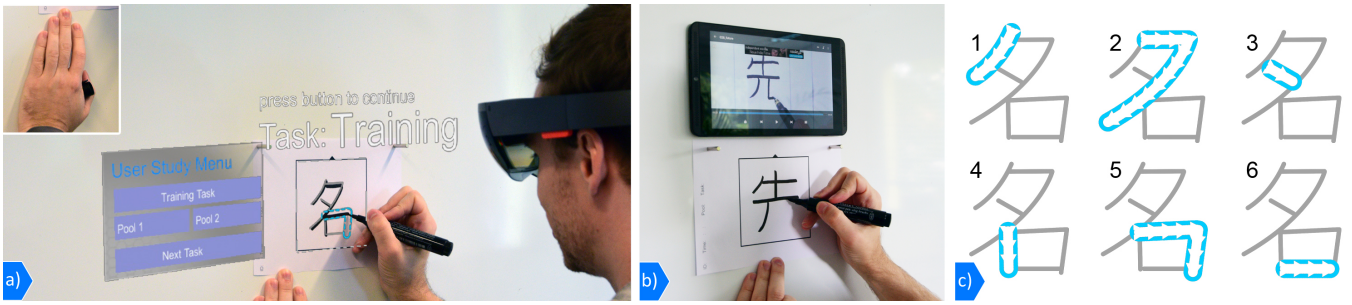

Figure 7. Retargeted Kanji tutorial and final revision of AR visualization. (a) The AR visualization is presented using an Optical See-Through HMD (Microsoft Hololens) and a handheld clicker that the user is holding in one hand. (b) The video tutorial is shown on a tablet mounted right above the drawing area. This reduced the influence of head motion. (c) Our final glyph design encodes the direction of the stroke on its border using arrow heads. The system presents one glyph at a time next to a full preview of the final drawing. This picture shows the six instructions presented to the user in AR.

Task completion time was measured using a stop watch from the point in time where participants announced the start of the drawing. The drawing was considered finished when the last stroke was placed. The error was measured as in the previous study using the l2-distance between the image pixels of the tutorial Kanji and user-drawn Kanji. **Hypotheses.** Due to the presentation of the tutorial using AR and the preceding authoring step to process the tutorial instructions, we expect that when working with AR2 users will be significantly faster and more accurate than when using unmodified video instructions (**H1**). Furthermore, users will prefer AR2, due to its improved usability and intuitive visualization (**H2**).

Results. 12 participants (3 female, $\overline{X}$ =31.3 (sd=6.2) years old) volunteered for the study. On a scale from one to five, five meaning best, the mean of self-rated AR experience was 2.7 (sd=1.2), video tutorial experience was 3 (sd=0.6), Kanji experience was 1.3 (sd=0.65) and general drawing ability rated as 2 (sd=1.3). With 12 participants, two interfaces and ten different symbols per interface, there were a total of $12 \times 2 \times 10 = 240$ trials. The data was evaluated using a level of significance of 0.05 and Wilcoxon signed rank tests.

Table 2 shows the mean, standard deviation and median of task completion time, error, NASA TLX and SUS for AR2 and VR2. Figure 8 shows the boxplots of the measurements. Wilcoxon signed rank tests revealed a statistically significant difference in task completion time (($Z = -3.0594, p < 0.001, r = 0.62$)), error ($Z = -2.9025, p < 0.05, r = 0.59$), NASA TLX ($Z = -3.0594, p < 0.001, r = 0.62$) and SUS ($Z = 2.3552, p < 0.05, r = 0.48$). In all cases AR2 outperformed V2. These results support H1 and H2.

All 12 participants preferred AR2 over V2, when asked to choose one of the two interfaces in the after-study questionnaire.

Discussion. Our results support H1 and H2. Our system clearly outperforms traditional video tutorials and is also preferred by the participants. The median SUS value of 92.5 for the AR interface is higher than the average of 70 and, based on the analysis of Bangor et al. [4], can be translated into the adjective "excellent". The traditional video interface has median SUS value of 76.25 and receives the adjective "good" [4].

Participants were very positive about the AR interface in the after-study interview and clearly preferred this interface, similar to the results of the previous study.

Five participants mentioned the clear benefit of AR in the ability to control the speed in which the instructions were shown. The lack of speed control in V2 was considered as stressful, because the video was either too slow or too fast. Two participants remarked that they appreciated the preview of the finished Kanji in the beginning of the instructions, which underlines the usefulness of our authoring system in reformatting video tutorials into a more sophisticated format for learning.

Four participants again noted problems with occlusions in the AR condition. The visualization sometimes occluded the drawn line, especially when they drew out of the bounds. While closing off the indicated drawing area could be regarded as desirable feature, it would be better, if a visualization could identify user-relevant content to avoid occlusion. Seven participants noticed that the focal plane of the Hololens did not match the surface they were working on. This is a long-known problem of this kind of HMD technology. While unpleasant, participants could still easily finish the task.

In condition V2, participants paused the video frequently to keep up with the instructions. This indicates the value of step-by-step instructions as extracted by our system. While we compared our system to a standard video interface V2, it would be interesting to compare the AR condition with a more advanced video interface that also supports step-by-step instructions, or even automatic pause-and-play techniques such as the one presented by Pongnumkul et al. [40]. We speculate that part of the performance difference between AR and V2 comes from the segmentation of the tutorials into steps in the AR condition and that the performance of a more advanced video tutorial will be closer to the AR condition.

Based on a visual comparison of the resulting drawings, we noticed no difference in jitter between drawings generated with interface V2 compared to interface AR2 (see the complementary material for scans of the drawings). Since we changed both, the glyph design and the application (to one which does not require deformable object modelling and tracking), a future experiment will have to investigate the actual impact of each of the two factors to jittery drawings in AR instructions.

Table 2. Measurements of Kanji study (mean (sd), median).

Cond.	Time (s)	Error	SUS	Nasa TLX
AR2	20 (3.7), 19.9	0.41 (0.03), 0.4	89.6 (10.1), 92.5	22.4 (11.6), 21.3
V2	30.7 (6.6), 30.1	0.46 (0.02), 0.46	68.1 (20.8), 76.3	46.5 (16.1), 45.4

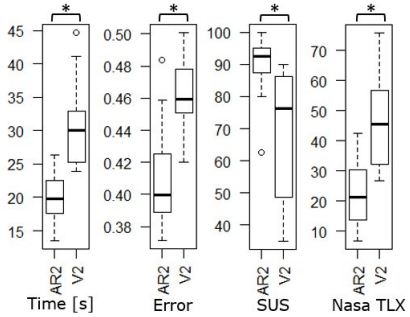

Figure 8. Kanji study results. Stars indicate significant differences.

DISCUSSION AND FUTURE WORK

We have presented a flexible framework for retargeting tutorials from video to AR. Users can quickly extract motions from video sources by providing just enough cues to initialize the motion reconstruction. The input required by the user is small, making the authoring process simple and swift.

We performed a series of evaluations to improve the design of our authoring system as well as the tutoring system. In particular, we have redesigned the AR visualization twice and the method to step through instructions once. In addition, we have extended the initial set of authoring tools based on user feedback. The resulting AR tutorial clearly outperformed video based instructions for precise drawing tasks.

Next to the revisions we have implemented, we noticed a number of challenging situations for our system. We compiled a number of recommendations to address challenges when systems for applications similar to ours.

Compensate for limited tracking during extraction. Any system following our approach to guide extraction based on feature tracking will be limited by the capabilities of current tracking methods. For example, if the input video is very dark, noisy or blurry, the surface and tool tracker may require excessive user input to manually correct failure cases. Similarly, subtle visual additions or color alterations, such as in facial make-up, cannot be detected with current computer vision techniques. While better cameras or improved multi-channel tracking methods can overcome these restrictions in the future, we recommend also using common 2D image drawing tools to provide the information manually. This will lead to a more time consuming authoring process, but allows by-passing difficult input material.

Compensate for limited tracking and scene modelling during playback. Our 3D face tracker is not able to precisely handle face deformations in 3D during playback. Our make-up experiment indicates that small errors in tracking and scene modelling has a high impact on the performance of the AR visualization. While better tracking technology can overcome this problem in the future, we recommend adding visual feedback for users, so that they can control their movements to avoid situations, where the tracking system will introduce large errors or potentially fail. For instance, the user can apply the feedback to stabilize deformable scenes and thus will avoid motion which may lead to floating augmentations.

Provide real-time feedback on the user's performance. Our current approach presents the trajectory and the direction of the motion the user has to perform. We do not provide any feedback on the performance of the user or other forms of guidance, while using the AR tutorial. Instead, we rely on the user's ability to directly observe any deviations in-situ. However, this requires additional user attention and may result in a stop and go strategy similar to what was observed in our make-up study. Obviously, with robust difference detection in real time performed on the user's video stream, we can extend the tutorial system to respond to the user's actions and provide feedback in a more explicit way, similar to the feedback proposed by Bau and Macay [5]. To limit the amount of visual information provided to the user, we furtermore recommend using other modalities, such as vibration or sound, to provide feedback.

Automatic contrast adaptation. Our system presents the extracted motion using the revised glyph design in a user defined color. This approach assumes similar contrast between real world background and the augmentation over the entire tutorial. However, this assumption limits the set of tutorials the system can effectively display. To ensure visibility of AR instructions at any point in time, adaptive approaches, such as saliency differencing [23], or static methods which provide contrast on the border of glyphs [22] should be considered.

Combine with video presentation Our work demonstrates how AR can support following video tutorials. However, we do not believe that AR should replace 2D video tutorials entirely. From our experience, it is most suitable for delicate motions but likely not as efficient as video to communicate coarse actions, such as filling an area. For more extensive procedures, the most powerful approach may combine conventional video and AR. For example, surface-filling actions do not require very precise motion. Only the border has to be handled carefully, while filling the interior does not require special motions. In such cases, conventional video can be interleaved with AR, depending on the specific requirements of the action.

Besides our design recommendations, several directions for future work exist. For example, we aim at increasing the number of object classes our system can support, such as assembly or dance tutorials. This requires human body motion tracking as well as tracking of other body parts, such as hands, from 2D video. Furthermore, we will investigate tools to effectively extract and visualize 3D tutorials which require precise motion in time. This will require the design of visualizations which encode speed, velocity and the direction of tools in 3D. An important part of this work will be the evaluation of these visualizations, which need to convey different attributes without distracting the user.

ACKNOWLEDGMENTS

This work was funded by a grant from the Competence Centers for Excellent Technologies (COMET) 843272 and the EU FP7 project MAGELLAN (ICT-FP7-611526).

REFERENCES

1. Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. 2003. Designing effective step-by-step assembly instructions. *ACM Trans. Graph.* 22, 3 (July 2003), 828–837.
2. Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: Enhancing Movement Training with an Augmented Reality Mirror. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 311–320.
3. Tadas Baltrušaitis, Louis-Philippe Morency, and Peter Robinson. 2013. Constrained Local Neural Fields for robust facial landmark detection in the wild. In *300 Faces in-the-wild challenge, International Conference on Computer Vision*.
4. Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies* 4, 3 (2009), 114–123.
5. Olivier Bau and Wendy E. Mackay. 2008. OctoPocus: A Dynamic Guide for Learning Gesture-based Command Sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 37–46.
6. P. Breedveld. 1997. Observation, Manipulation, and Eye-Hand Coordination Problems in Minimally Invasive Surgery. In *in Proc XVI European Annual Conference on Human Decision Making and Manual Control*. 9–11.
7. John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
8. Andreas Butz. 1994. BETTY: Planning and Generating Animations for the Visualization of Movements and Spatial Relations. In *Proc. of Advanced Visual Interfaces*. 53–58.
9. Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 93–102.
10. Pei-Yu (Peggy) Chi, Mira Dontcheva, Li Li Wilmot, Daniel Vodel, and Björn Hartmann. 2016. Authoring Illustrations of Human Movements by Iterative Physical Demonstration. In *Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology (UIST '16)*. to appear.
11. Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. 2002. Video Matting of Complex Scenes. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*. ACM, New York, NY, USA, 243–248.
12. J P Collomosse. 2003. Cartoon-style Rendering of Motion from Video. *Vision Video and Graphics* 67, 6 (2003), 549–564.
13. Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. 2014. You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video. In *British Machine Vision Conference (BMVC)*. BMVA.
14. David H. Douglas and Thomas K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 2 (1 Oct. 1973), 112–122.
15. Steven Feiner, Blair Macintyre, and Dorée Seligmann. 1993. Knowledge-based augmented reality. *Commun. ACM* 36 (July 1993), 53–62. Issue 7.
16. M. Goto, Y. Uematsu, H. Saito, S. Senda, and A. Iketani. 2010. Task support system by displaying instructional video onto AR workspace. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*. 83–90.
17. Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 66.
18. Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. 2012. DuploTrack: A Real-time System for Authoring and Guiding Duplo Block Assembly. In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST '12)*. 389–402.
19. Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology* 52 (1988), 139–183.
20. Steven Henderson and Steven Feiner. 2011. Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair. *IEEE Trans. Vis. Comp. Graph.* 17, 10 (2011), 1355–1368.
21. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2012. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 7 (July 2012), 1409–1422.
22. Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. 2009. Explosion Diagrams in Augmented Reality. In *Proc. of IEEE Virtual Reality (VR '09)*. IEEE, 71–78.
23. Denis Kalkofen, Eduardo E. Veas, Stefanie Zollmann, Markus Steinberger, and Dieter Schmalstieg. 2013. Adaptive ghosted views for Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013, Adelaide, Australia, October 1-4, 2013*. IEEE, 1–9.

24. Bernhard Kerbl, Denis Kalkofen, Markus Steinberger, and Dieter Schmalstieg. 2015. Interactive Disassembly Planning for Complex Objects. *Computer Graphics Forum* (2015).
25. B. Kim and I. Essa. 2005. Video-based nonphotorealistic and expressive illustration of motion. In *Proc. of the Computer Graphics International 2005*. 32–35.
26. Vladislav Kraevoy, Alla Sheffer, and Michiel van de Panne. 2009. Modeling from Contour Drawings. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling (SBIM '09)*. ACM, New York, NY, USA, 37–44.
27. Tobias Langlotz, Mathäus Zingerle, Raphael Grasset, Hannes Kaufmann, and Gerhard Reitmayr. 2012. AR Record Replay: Situated Compositing of Video Content in Mobile Augmented Reality. In *Proceedings of the 24th Australian Computer-Human Interaction Conference (OzCHI '12)*. 318–326.
28. Florian Ledermann and Dieter Schmalstieg. 2005. APRIL A High-Level Framework for Creating Augmented Reality Presentations. In *Proc. of IEEE Virtual Reality*. 187–194.
29. V. Lepetit, F.Moreno-Noguer, and P.Fua. 2009. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal Computer Vision* 81, 2 (2009).
30. Wilmot Li, Maneesh Agrawala, Brian Curless, and David Salesin. 2008. Automated Generation of Interactive 3D Exploded View Diagrams. *ACM Trans. Graph.* 27, 3, Article 101 (Aug. 2008), 7 pages.
31. David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision* 60, 2 (Nov. 2004), 91–110.
32. Bruce D. Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2 (IJCAI'81)*. 674–679.
33. Peter Mohr, Bernhard Kerbl, Michael Donoser, Dieter Schmalstieg, and Denis Kalkofen. 2015. Retargeting Technical Documentation to Augmented Reality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3337–3346.
34. G. Nebehay. 2012. *Robust Object Tracking Based on Tracking-Learning-Detection*. Master's thesis. Faculty of Informatics, TU Vienna.
35. Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11)*. IEEE Computer Society, Washington, DC, USA, 127–136.
36. Marc Nienhaus and Jürgen Döllner. 2003. Dynamic glyphs - depicting dynamics in images of 3D scenes. In *Proc. of the 3rd international conference on Smart graphics (SG'03)*. Springer-Verlag, Berlin, Heidelberg, 102–111.
37. Peter Ondruska, Pushmeet Kohli, and Shahram Izadi. 2015. MobileFusion: Real-time Volumetric Surface Reconstruction and Dense Tracking On Mobile Phones. In *International Symposium on Mixed and Augmented Reality (ISMAR)*. Fukuoka, Japan.
38. M. Park, S. Serefoglou, L. Schmidt, K. Radermacher, C. Schlick, and H. Luczak. 2008. Hand-Eye Coordination Using a Video See-Through Augmented Reality System. *The Ergonomics Open Journal* 1 (2008), 46–53.
39. N. Petersen and D. Stricker. 2012. Learning task structure from video examples for workflow tracking and authoring. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*. 237–246.
40. Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-play: Automatically Linking Screencast Video Tutorials with Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 135–144.
41. Dorée Duncan Seligmann and Steven Feiner. 1991. Automated Generation of Intent-Based 3D Illustrations. In *Proc. of ACM SIGGRAPH*. 123–132.
42. Jianchao Tan, Marek Dvorožňák, Daniel Sýkora, and Yotam Gingold. 2015. Decomposing Time-Lapse Paintings into Layers. *ACM Transactions on Graphics (TOG)* 34, 4, Article 61 (July 2015), 10 pages.
43. Richard Tang, Xing-Dong Yang, Scott Bateman, Joaquim Jorge, and Anthony Tang. 2015. Physio@Home: Exploring Visual Guidance and Feedback Techniques for Physiotherapy Exercises. In *Proceedings of ACM CHI*. 4123–4132.
44. Barbara Tversky, Julie Bauer Morrison Y, and Mireille Betrancourt. 2002. Animation: Can it facilitate. *International Journal of Human-Computer Studies* 57 (2002), 247–262.
45. Sean White, David Feng, and Steven Feiner. 2009. Interaction and presentation techniques for shake menus in tangible augmented reality. In *Proc. of the IEEE IISMAR*. 39–48.
46. Jürgen Zauner, Michael Haller, Alexander Brandl, and Werner Hartmann. 2003. Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures. In *Proc. of IEEE/ACM ISMAR*. 237–246.