AssemblyMixT: Designing Mixed-Media Assembly Instructions

Pei-Yu (Peggy) Chi
Computer Science Division, UC Berkeley
peggychi@cs.berkeley.edu

ABSTRACT
Following assembly instructions can be challenging, especially for complicated tasks such as composing furniture pieces and 3D printed objects. Followers need to interpret the instructions, match with real-world pieces, and transfer the knowledge into physical actions to manipulate the objects. In this paper, we design a new, interactive navigation UI that combines static, step-by-step illustrations and video instructions from a collection of furniture assembly instructions produced by a professional crew. Computer vision methods are applied to automatically segment an instructional video and match with the corresponding static instructions.

Author Keywords
video; instructions; tutorials; demonstrations; how-to

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Assembly instructions show how to perform step-by-step operations for building up everyday products from parts in a procedural way. Useful information includes the parts (e.g. bolts, wheels, and boards) and their orientation (e.g. upside down), and the actions involved (e.g. screwing, sliding, and snapping). However, following instructions can be challenging, especially for complicated tasks such as composing furniture pieces, electronic devices, and 3D printed objects. Followers need to interpret the instructions, match with real-world pieces, and transfer the knowledge into physical actions to manipulate the objects.

There are two popular formats of assembly instructions: Static instructions, often shown as a step-by-step flow, effectively depict the important parts and actions. Common visualization techniques include ordering and orienting the parts, showing guidelines and arrows to depict an operation, and adding callouts for details [2]. 3D models are often drawn as illustrations, presented by lines and selectively with shades, textures, and colors to visualize the objects. Video instructions, on the other hand, demonstrate the exact actions of a person performing the task. To preserve only necessary details, video editing techniques including cuts, transitions, zooms, picture-in-picture views, and visual highlights, can be applied [6].

The two formats can be effective in different ways: Static instructions provide condensed, abstract information that can be viewed at a glance, but they might require viewers’ efforts to interpret and put into the exact actions. Video instructions capture the actual views of the parts and useful details of physically assembling a product, but they might be hard to navigate with a conventional video player [5].

In this paper, we explore the space between static and video instructions and understand their benefits. In particular, we focus on the visualization techniques and design a new, interactive navigation UI to support a task-following scenario. To help compare the effects, professional instructions created by IKEA are used as supportive material of the system. Computer vision methods are applied to automatically segment an instructional video and match with the corresponding step-by-step illustrations.

The main contributions of this paper include:
• A computer vision algorithm to detect steps, shots, and editing effects (including transitions, closeups, and illustrations) in a professionally edited instructional video that matches with step-by-step illustrations.
• UI designs for viewing mixed-media assembly instructions that include illustrations, screenshots, and videos.
• A working implementation of this approach and a preliminary evaluation.

The following paper is structured as follows: We survey the related work and common techniques used in creating technical illustrations. We introduce our system and UI design, followed by an automatic shot detection algorithm for mapping professional videos and static illustrations. We present the results and a preliminary evaluation to discuss our approach.

RELATED WORK
Automatic Generation of Instructions
Creating instructions can be a time-consuming process. Researchers have been studying automatic methods to generate instructions. This often requires identifying visualization principles and carefully designing automation techniques.
Based on a study of understanding how people create and follow furniture assembly instructions [15], Agrawala et al. designed a system that plans and presents a step-by-step diagram considering the geometry, orientation, and ordering constraints of a 3D object [2]. Approaches to automatically create exploded view diagrams [18, 20] and a causal chain sequence of how mechanical assemblies move [19] demonstrate different methods of visualizations based on 3D model understanding. Chi et al. surveyed and interviewed with tutorial authors to define a set of common video editing techniques. This was applied to design a semi-automatic system to create instructional videos [6]. There exists a great amount of work in the software application domain including MixT, an automatic approach of creating mixed-media tutorials using screen cast videos and application logs [5]. However, it is beyond the scope of this paper and will not be discussed here.

Providing Real-time Instructions
By recognizing physical activities, technologies can support users by providing real-time instructions. DuploTrack uses a depth camera to track progress and provide visual guidance for block assembly tasks [13]. Information can also be augmented in the real world through a wearable display for maintenance work [16, 17] or using a projector for remote collaboration [9, 14].

ILLUSTRATION TECHNIQUES
To design mixed-media assembly instructions, it’s worth understanding the design principles and visualization techniques used in technical illustration, which is to communicate the structure and details of a technical or engineering object.

Representations
One of the basic representations is line drawing, which consists of a set of straight or curved lines to visualize an object without shading or color (Figure 1A1). Early studies have developed automatic techniques to create line illustrations [8, 7]. Other approaches can be used for different purposes, including showing various parts at once for a complex object or presenting the inner structure [1, 10]:

- Exploded view shows the ordering or relative positions of parts (Figure 1B2). Often parts are labeled with text and guidelines.
- Cutaway shows the internal parts of an object by removing outer features (Figure 1C3). This helps to provide an overall context while showing the assembled details to “see through” an object.
- Ghosting uses transparency to show different levels of details (Figure 1D4). Instead of completely removing the parts like a cutaway, this method preserves the clarity.

1http://www.crateandbarrel.com/Assembly-Instructions/English/Felix_Chair.pdf
2http://www.rjc-technical.co.uk/Pages/black-decker-drill.htm
3http://www.lineaforma.com/illustration2.html
4http://www.doschdesign.com/products/3d/Car_Details_V2.html

Figure 1. Examples of illustration types: A) Step-by-step line drawing, B) Exploded view, C) Cutaway, and D) Ghost representation.
**Perspectives**
Illustrations are commonly shown using the parallel perspective, which is efficient to present the spatial details of an object. True perspective considers a vanishing point in a drawing, similar to how we perceive an object. However, it’s relatively difficult to create and reuse the illustrations and therefore is less common.

**Styles**
Styles need to be carefully designed to depict 3D objects in a consistent way [10]:
- Variables including line weights (thick or thin), line types (solid or dashed), and brightness or color are used to infer the features.
- Magnifier and callout help show the details in a larger scale. The detailed view can be enclosed by a circle, an oval, or a square, connected by lines or arrows to the main illustration.

**Shading and Texture**
To provide more depth and to show the material of an object, shading and texture are often applied to enhance drawings. Previous work has presented automatic techniques using image-based relighting [3], line direction [11], principle directions [22], and lit sphere [25]. Masks and visual patterns can be used to create texture, such as metallic, crisp, or flurry material. For example, the shading pattern in Figure 1C differentiates spheres and cavities.

**Discussion and Opportunities**
The goal of designing instructions is to help viewers perceive and understand a concept while providing necessary details. In a visualization process, it is always a trade-off to keep or remove information, to visualize in a more abstract or realistic way, and to remain simple or add levels of details.

We suspect that an interactive visualization interface would help enhance the visual design that shows details based on viewers’ needs. For example, videos may serve as the extreme detailed view, while line illustrations provide a basic view. The challenge, therefore, lies in balancing between the trade-offs and the amount of interaction.

**DESIGNING MIXED-MEDIA ASSEMBLY INSTRUCTIONS**
We introduce AssemblyMixT, an interactive user interface for navigating assembly instructions with mixed-media materials. Our system analyzes a professionally edited instructional video and matches with the corresponding step-by-step illustrations. It provides five viewing modes to navigate the instructions, including a list view, a shot-list view, a carousel view, a step-centric view, and a video view.

**Material**
To design mixed-media tutorials with both static and video instructions, we collect furniture assembly examples that include step-by-step procedures at certain degree of complexity. In order to ensure that both formats provide the same or similar quality, we surveyed professional instructions on-line and found that IKEA’s recently produces a set of well-edited videos that introduce the building processes of some of their products as the support of printed instructions. Figure 2 shows one example “KIVIK Sofa” and its static and video instructions of one step.

**Styles**
We analyzed the styles used in the collected instructions for designing automatic detection algorithm and new user interface. Below shows a list of the visualization techniques used in the IKEA static instructions:
- Illustrations: Products and parts are presented as line illustrations in black and white. Shading is used for soft material such as a blanket.
- Final product: Instructions start with a view of the assembled product and the product name.
- Parts: A part list shows the amounts (e.g. “12x”) of the parts required and the part numbers next to the drawings.
- Steps: Instructions are shown in a list of steps, starting from step number 1.

---

5 [http://www.ikea.com/us/]
7 [http://www.youtube.com/watch?v=cigz25_vpgU]
• **Orientation**: Parts are oriented to inform how they should be positioned.

• **Focus+Context**: Callouts in a round circle or a square are effective to show the detailed views, especially for smaller parts such as screwing a bolt (e.g. the two circles in Figure 2). The number of the same operation is labeled next to the callouts when necessary, such as “4x”.

• **Alignment**: Solid or dashed lines depict how parts should be aligned (e.g. thin lines in the two callouts of Figure 2).

• **Actions**: Arrows indicate actions, such as to screw in a clockwise or anti-clockwise direction (the circular arrows) or to push a part (the straight arrow in Figure 2).

• **Human figures or hand** are sometimes drawn to show the suggested positions to stand or hold (e.g. Figure 2).

The corresponding IKEA “How To Build” videos are produced with a consistent style as the static instructions but apply several video editing techniques:

• **Mixed style**: Videos include illustrations, demonstrations with actors, visual enhancements, and animation.

• **Final product**: Videos start with an actual image of the final product and the product name.

• **Steps** are presented in a temporal order. Step numbers are shown on the top-left corner of each shot.

• **Parts and illustrations**: Each step starts with a highlighted part list and an illustration from the static instructions.

• **Shots**: Each step includes one or more shots showing one or two actors who demonstrate how to assemble the parts. A shot can be an overview, a picture-in-picture view, or a closeup (Figure 3). Most shots are connected by jump cuts, except that a transition is applied from the static illustration to a video shot.

• **Highlights**: A shot may be highlighted by animated arrows or a check or cross marker to point out the important parts. Text annotations are often applied (Figure 3 bottom).

### User Interface Designs

Our goal is to automatically identify the video shots and match with static instructions. In this way, we can design a user interface for viewers to interactively navigate the step-by-step instructions with both static and video details:

#### List View

Inspired by how a printed manual presents the instructions in a linear step-by-step sequence, we embed the corresponding video clip side-by-side with each static illustration (Figure 4A). Viewers can quickly scan through the illustrations on a web page. When more details are needed, viewers can click on the illustration and see the video clip. To help viewers identify the connection between two formats, the system tracks the current shot being played and highlights the illustration for any closeup view using a red circle (Figure 5). In order to avoid redundant information provided by both static and video instructions, all the illustration shots in a video are purposely skipped based on the metadata from our shot detection algorithm.

#### Shot-list View

It’s possible that a representative frame in a shot provides sufficient information than playing a video clip or showing animation. Therefore, the shot-list view presents all the video shots next to the illustration for each step (Figure 4B) so that viewers can glance through the process.

#### Carousel View

Instead of presenting all the shots as a list, the carousel view shows only one representative frame at a time like a slide show.

#### Step-centric View

Similar to a video player that only shows one step at a time, in the step-centric view, only one illustration is shown with the video clip next to it (Figure 4C).

#### Video View

Finally, the video view shows only the original instructional video without listing the static illustrations next to it.

### AUTOMATIC EFFECT DECISION PIPELINE

Based on the analysis of the video editing decisions that go into the IKEA instructional videos, we design a shot detection algorithm using computer vision techniques. Our system first detects video shots and identifies the editing effects:

#### Shot Detection

A shot, defined as “a series of frames that runs for an uninterrupted period of time” [24], is the basic unit of a video that we are interested in. A shot might be static (such as showing a step illustration for 3 seconds) or non-static (when an actor
performs an action such as screwing a bolt). We observed that there are three types of a shot change:

- **Consecutive illustration shots**: The video shows two different illustrations in a sequence, such as from a part list to a step illustration. The frames have similar or identical dominant colors (in this case, light blue in the background and white for the lines, see Figure 3), but the line illustrations are different.

- **Shots with an abrupt change**: such as from a step illustration to a demonstration shot with actor(s) and the furniture. The visual content changes significantly in both colors and lines.

- **Consecutive video shots**: such as from an overview shot of an actor and the furniture to a closeup shot that shows the actor’s hand. The scene remains unchanged with the same objects and actor(s), but the views are changed between an overview and a zoomed region.

We did not find any shot with camera movement (e.g., pan from the left to the right). In order to detect shot changes in a video considering the above cases, we compare both the edge difference and the pixel histogram between every two consecutive frames.

1. Given an input video $V$ of duration $d$ (in seconds) and frame rate $fps$ (frames per second), retrieve its series of frames $[f_1, f_n]$, where the total number of frames $n = d \times fps$.
2. For each true-color frame $f_i$ in RGB space, find the edges of objects in blocks as $e_i$ using Sobel edge detection algorithm. Convert $f_i$ to the grayscale intensity image and compute its normalized histogram in 256 bins as $h_i$. 

---

Figure 4. Viewing modes of the AssemblyMixT user interface: A) List view, B) Shot-list view, and C) Step-centric view.

Figure 5. Highlight on the fly to enhance a closeup view.
3. Compute the edge difference $E_{i,i+1}$ between $f_i$ and $f_{i+1}$ by getting the absolute block difference between $e_i$ and $e_{i+1}$.

4. Compute the histogram difference $H_{i,i+1}$ between $f_i$ and $f_{i+1}$ by getting the Euclidean distance of each pair between bins in $h_i$ and $h_{i+1}$.

5. Define a shot $shot_x$ is changed when 1) the count of $E_{i,i+1} > \varepsilon_{edge}$ (set as 10%) is over a high threshold $\varepsilon_{high}$, or 2) the count of is over a lower threshold $\varepsilon_{low}$ and $H_{i,i+1}$ is over a threshold $\varepsilon_{hist}$ (set as 10%). Set the start time $st_x$ of the shot as $\frac{i+1}{fps}$ (in seconds).

6. Merge two consecutive shots $shot_x$ and $shot_{x+1}$ if $st_{x+1} - st_x$ is within a threshold of minimum shot length $min_t$ (set as 1.5 seconds).

In this way, we can detect each shot change and record the start and end time of each shot as well as their lengths. We pick the middle frame as the representative image of a shot.

**Shot Effects**

To determine the shot type and effect of a shot, we follow the below heuristics:

- **Illustration**: If the major color (>90%) of the representative frame is blue, label this shot as an illustration. The frame is converted to the HSV color space in order to identify the range of color blue (Hue [0.4, 0.7], Saturation [0.2, 1], and Value [0.5, 1]).

- **Fade**: When a shot is merged by consecutive shots within the time threshold $min_t$, label this shot has a fade transition.

- **Static**: If both the edge and histogram differences in a shot are under a very low threshold, label this shot as static.

- **Closeup**: For every frame in the video $V$, find all the circles of radius ranged from 10-20% of the video width in a frame using circular Hough transform [26]. If a frame in a shot includes one or more circles, label this shot as a closeup view. Record the $x, y$ position and radius $r$ of each circle for visualization.

**Step Grouping**

Finally, we identify the step number of each shot by template matching using an image repository. Our system assumes that any shot between two shots of the same step number should be grouped into one step. Therefore, we take this as a string completion problem and label each intermediate shot with a step number except for the introduction at the beginning (i.e. before step 1) and the end. We record the start and end time and the total length of each step.

**Analyzing Static Illustrations**

We also analyze the static illustrations using the same technique of finding a closeup view. We detect circular shapes and record every $x, y$ position and radius $r$. In this way, when a video shows a closeup shot, the UI highlights the corresponding magnifier in the illustration.
false alarm (incorrect shots), and fn is false negative or a miss (missed shots). An interactive visualization UI using d3 [4] is designed to help the experimenters examine the shot detection results (Figure 7). Our system found about 90% of shots and worked especially well (100% for both precision and recall) for two videos. Only few shots (1.25 among 57.75 shots on average) were incorrectly labeled as positive. The system fails to detect shots when the changes are minor, such as switching a camera angle of an overview scene.

In general, this result shows that our shot detection algorithm is able to automatically segment a video in order to generate mixed-media tutorials.

Figure 6 shows examples of the closeups detected by our system in both illustrations and video frames. This accurate information is useful for the user interface to dynamically highlight the views (In the UI we purposely expand the radius by 20 pixels to cover the whole closeup, see Figure 4A).

**User Evaluation**

The goal of designing AssemblyMixT is to assist viewers who follow and assemble objects in practice. Therefore, we collect user feedback to evaluate the usability and utility of our system in the following ways:

**Instruction Following**

The author and one partner tested with an earlier version of the UI to follow the instructions of “MALM Bed Frame”. Printed material was not provided. The viewers first watched the video using the UI. They then followed instructions using the list mode to assemble the bed frame in 30 minutes. When assembling the furniture, they constantly viewed the static illustrations back and forth with an iPad to confirm the progress. They did not watch the video for a certain step but explained that for an unfamiliar or complicated task, video might be efficient and helpful to quickly review the details before proceeding the next step.

**Preference of Viewing Modes**

To understand viewers’ preference of viewing modes, we collected feedback in a class poster session. The audience was interested to learn the concept of mixed-media instructions and to see how a conventional step-by-step tutorial can be “animated” and interactively navigated. Some participants were curious to know the opportunities of automatic navigation while assembling furniture pieces. Real-time feedback shown on a head-mounted display such as Google Glass was also mentioned. An experiment studying how followers would want to interact with the media would be interesting.

**LIMITATIONS AND FUTURE WORK**

The current shot detection algorithm is designed specifically for the IKEA-style videos. Segmentation using edge and histogram differences can be applied to general videos, but the techniques of finding steps and closeups will need to be adjusted for other video styles.

Our system analyzes and combines existing material (illustrations and videos), but it might be interesting to generate different levels of details such as automatically rasterizing video frames to create realistic illustrations. It’s also possible that pure animation without human actors and real-world objects is sufficient, such as the example shown in http://www.youtube.com/watch?v=37v96zMllFA. Also, visual designs that effectively utilize the screen estate should be considered. Previous work has shown video tapestries and storyboard [12] and synopsis [21] are effective to visualize activities in a video.

The larger scope of this paper is to support capturing and generating mixed-media tutorials. Studies surveyed the hack culture of amateurs who love to build and share their assembling experiences [23]. One challenge would be to support users documenting the process into an edited format for the community to view and participate.

**CONCLUSION**

In this paper, we introduced a new, interactive navigation UI that combines both static, step-by-step illustrations and videos from a collection of furniture assembly instructions produced by a professional crew. Computer vision techniques to automatically segment an instructional video were introduced. Several viewing modes and examples were presented. All in all, this project opens several opportunities of capturing and consuming assembly instructions interactively.

**REFERENCES**


4. Bostock, M., Ogievetsky, V., and Heer, J. D: Data-Driven Documents. IEEE Transactions on
Figure 7. Interface that visualizes the shot detection results for measuring the accuracy (Edge difference on the top and histogram difference on the bottom. The shot effects were shown as colored markers.)


