In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), 2008

Spatiotemporal Coding in ANVIL

Michael Kipp
DFKI, Embodied Agents Research Group
Campus D3.2, 66123 Saarbrücken, Germany
E-mail: michael.kipp@dfki.de

Abstract

We present a new coding mechanism, spatiotemporal coding, that allows coders to annotate points and regions in the video frame by drawing directly on the screen. Coders can not only attach labels to time intervals in the video but can specify a possibly moving region on the video screen. This opens up the spatial dimension for multi-track video coding and is an essential asset in almost every area of video coding, e.g. gesture coding, facial expression coding, encoding semantics for information retrieval etc. We discuss conceptual variants, design decisions and the relation to the MPEG-7 standard and tools.

1. Introduction

ANVIL [1] is a free video annotation research tool for adding structured human annotations to digital video material (Kipp, 2004). Relevant research areas where ANVIL is in active use include psychology, psycholinguistics, embodied conversational agents, human-computer interaction, computer vision, computer animation, anthropology, ethology and many others. The tool was designed for the efficient manual annotation of large video copora and is implemented in Java (using JMF[2]) for platform-independence, i.e. it runs on Windows, Linux and Mac machines.

In this paper, we present a new coding mechanism called spatiotemporal coding that allows coders to annotate points and regions in the video frame by drawing directly on the screen. This means that coders can not only attach labels to a certain time interval in the video, which is what annotation tools traditionally do, but can also specify a – possibly moving – location on the video screen. This quite literally opens up a new dimension for coding and where formerly coders could only refer to the complete video frame(s) they can now restrict the annotation to specific points or regions in the frame. This new class of coding is an essential asset in almost every area of video coding, e.g. gesture coding, facial expression coding, encoding location-based semantic data for information retrieval etc.

A number of tools similar to ANVIL have been developed in recent years[3] (cf. Rohlfing et al., 2006; Bigbee et al., 2001) and most of these tools share two key properties: (1) coding is performed along a horizontal timeline, i.e. time intervals are represented as horizontal bars and time points are points on this line, and (2) the coder has several of these lines (called tracks, tiers or layers) at his/her disposal to code different types of information. This has the important implication that all coding is fundamentally time-based. Each encoded entity (often called an annotation) is either attached to a time point or to an interval. The actual content of the annotation is usually a simple string, although ANVIL allows more complex structures to hold the information (Kipp, 2001).

It is important to be aware of the time-based nature of these tools in order to see how nontemporal data can be included in an elegant way. For instance, in (Martin & Kipp, 2002) we introduced nontemporal elements that refer to the video as a whole (instead of belonging to a certain time interval) and can thus encode persistent objects that occur in the video all the time or abstract objects that are only referred to by speech or gesture.

As for spatiotemporal coding, since ANVIL is a time-based tool, we extended it by allowing to *add* spatial annotations to *time-based elements*. We explain the implications, possibilities and restrictions of this particular design angle on spatiotemporal coding and report on the current state of this feature in the ANVIL tool. We start out by reviewing related tools with a special look at the MPEG-7 standard.

2. Related Work

A direct predecessor to this technology is an early ANVIL plugin called *Graphical Visual Markup*[4], developed by Christoph Lauer (DFKI) within the European NITE project. It allowed the coding of a fixed size moving region but had a difficult-to-use interface. Development has stopped and the plugin is no longer compatible with ANVIL's current version.

Adding structured information to a region (e.g., a bouding box) in an image or video frame is not a new idea. Especially in the context of the Semantic Web there has been high interest in marking-up multimedia data with "semantic" data on various levels of sophistication, from simple labels to ontological entities. This common interest culminated in the MPEG-7[5] standard (Martinez et al., 2002) for multimedia content description and a number of tools that support it.

[1] http://www.anvil-software.de
[2] Java Media Framework, http://java.sun.com/jmf
[3] See also http://www.ldc.upenn.edu/annotation/gesture
[4] see http://www.dfki.de/nite under "Anvil tools".
[5] http://www.chiariglione.org/MPEG/technologies/mp07-mds/index.htm

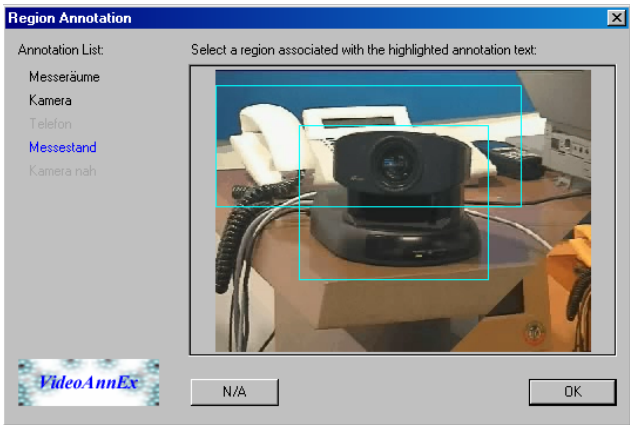


Figure 1: IBM's VideoAnnEx tool for MPEG-7 coding.

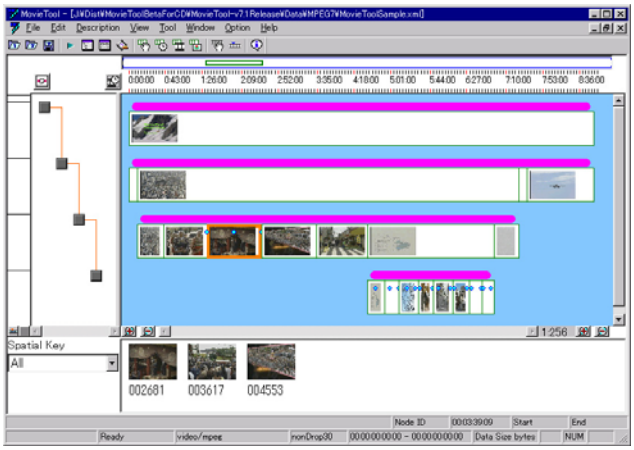


Figure 2: Ricoh's MovieTool allows temporal segmentation.

2.1 MPEG-7

MPEG-7 is an XML based description language for multimedia data, standardized under *ISO/IEC 15938: Multimedia Content Description Interface*. The basic elements are descriptors (D) and description schemes (DS). The description definition language (DDL) allows the definition of new DS's. Video segments are described in an entity called VisualSegmentDS which has a timestamp (TimeDS) and thus corresponds to a single frame. The GeometryDS allows to restrict the content to a specific region in the frame. Therefore, VisualSegmentDS is essentially a spatiotemporal descriptive entity.

However, having the capacity to define spatiotemporal annotations in XML does not suffice. One needs a tool to efficiently encode this data.

2.2 MPEG-7 Tools

VideoAnnEx was developed by IBM to allow MPEG-7 annotation of videos. The tool had a visual interface for coding a non-moving, rectangular region for a time segment (see Figure 1). However, according to the IBM website[6] the technology has been retired (entry dated July 19, 2002).

MovieTool[7] was a commercial MPEG-7 coding tool, developed by Ricoh. It allowed the manual (and semi-automatic) segmentation of the video in hierarchical segments (e.g. scenes). For these segments, the user could edit the corresponding MPEG-7 entry in a kind of augmented XML editor. Drawbacks are that the tool only read MPEG-1 coded videos and that there was no visual interface to encode screen regions. Also, the tool's development and distribution has stopped (website entry dated May 27, 2005).

M-OntoMat-Annotizer is a Java based tool for image/video annotation. Like ANVIL, it uses the Java Media Framework (JMF) for video playback. Single frames of a video can be coded using an MPEG-7 ontology (stored in RDFS). Regions can be defined manually or automatically (using visual feature extraction). However, the tool does not offer a timeline view where temporal segments are shown. Instead, it relies on a frame-by-frame view. Moreover, the tight integration with ontological tools and concepts makes usage more difficult, especially for users from non-technical fields.

3. Coding in Space and Time

In traditional annotation tools, the human coder adds annotation elements that, in the simplest case, carries a single label like "head nod" or "gaze right" or "beat gesture". This annotation element is usually further specified in two regards. First, the coder puts it into a certain track (aka tier or layer) which is something like a container, similar to a directory folder, that keeps all annotation elements of similar "type" together (for instance, one track for head movements, one track for gestures etc.). Second, the coder specifies a beginning and end time, restricting the element in time, which is usually visualized as a horizontal bar on a left-to-right timeline (see Figure 3).

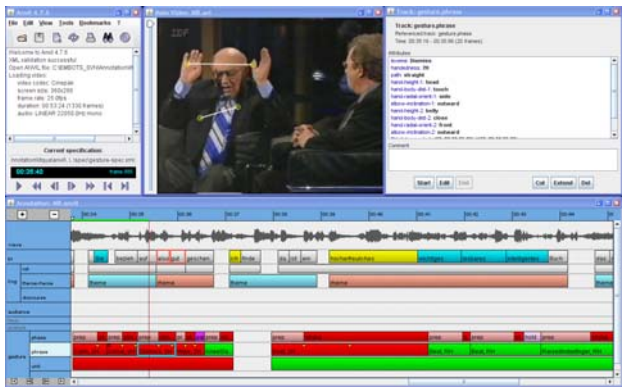


Figure 3: ANVIL video annotation tool.

Since there is no *spatial* restriction, the annotation element refers to the whole video frame (or rather the sequence of frames in that time interval). However, if the

[6] http://www.alphaworks.ibm.com/tech/videoannex
[7] http://www.ricoh.co.jp/src/multimedia/MovieTool

coder wants to specify that the annotation refers to, e.g., a building or a face in the upper left corner of the frame, this could be done by drawing a rectangle and storing its data. If this feature is moving (due to own movement or camera movement), the coder wants the annotation to correctly "follow" the feature for each frame of the video. In the next section we suggest a taxonomy of spatiotemporal coding. This could be done by specifying the location for each frame (time-consuming) or by specifying certain extreme or *key* locations and *interpolating* in between, like in computer animation.

Spatial annotation in ANVIL still uses time as the primary anchoring mechanism. This makes sense because video, like sound, is essentially a time-based medium. Exchanging this time-base for a spatial basis implies annotating a region on the video screen first and then restricting it in space. Having both time and space as equally important anchors would probably call for a 3D visualization as depicted in Figure 4, where annotation elements have three dimensions (time + 2D screen region). In the depicted example the region of each element neither moves nor changes shape.

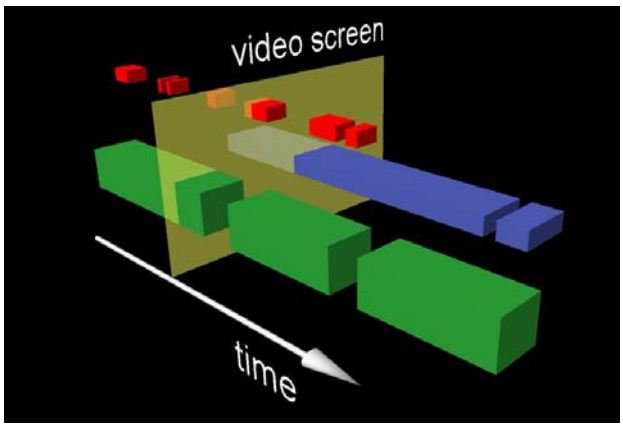


Figure 4: Coding time and 2D space at the same time could result for a full-fledged 3D interface as shown here. However, 3D interfaces require complex control mechanisms and would dampen the user's learning curve.

However, 3D interfaces have a number of drawbacks. They are difficult to control and are potentially visually misleading (occlusion, misinterpretation) since the user's screen is only two dimensional. Also, in a multi-tiered tool, every tier would each have to have such a 3D visualization. Finally, Figure 1 depicts only the simple case where regions do not move and do not morph. In the most complex case, these rectangular boxes would become amorphic worms meandering through space.

4. Taxonomy of Spatiotemporal Annotation

Having discarded the unifying 3D view, we keep the traditional video plus timeline view (Figures 3+5) as our primary visualization. To accommodate spatial data, each of the boxes in Figure 5 can contain spatial annotation, e.g. a list of points or a single static region. To clarify the various options for spatial representation, we distinguish the following dimensions: shape, number, ordering, rigidity and interpolation.

Shape means that the user has various geometric objects to specify the annotated region: a point, a rectangle, an ellipsoid, a polygon. Number means the user can either encode only a single region for the whole annotation element or a whole list/set. Ordering means that the various regions are either temporally ordered (i.e. for each region a timestamp is also registered) or have no paricular order. Rigidity means that a rigid region has a constant form whereas it is thinkable that a region varies in form (think of a camera zoom in/out). Interpolation refers to the possibility to offer linear or spline interpolation between the timestamped regions, allowing for economic coding of motion where only *key points* have to be coded. The taxonomy is illustrated in Table 1.

dimension	**values**
shape	point * rectangle ellipsoid polygon
number	single many *
order	unordered chronological *
rigidity	rigid * morphing
interpolation	discrete * linear * spline

Table 1: These dimensions specify a taxonomy of spatiotemporal coding, e.g. point-many-unordered-rigid-discrete. Asterisk means: available in Anvil.

5. User Interface

Any new feature to an annotation tool must be implemented on three layers. (1) It must have a file representation, (2) it must be represented and handled internally and (3) it must be made accessible through a (usually) graphical interface. (1) is important in terms of file interchange and the question of tools interoperability. (2) plays a role for plugin programmers who access internal data structures. Finally, (3) is the handle for the everyday user of the tool.

ANVIL distinguishes itself from most other coding tools in that it can hold several pieces of information in a single annotation element. Each element has so-called *attributes* that contain numbers, alphanumeric strings, a truth value etc. according to the attribute's *type*. The spatial data is stored in such an attribute. The *attribute type* specifies which kind of spatiotemporal annotation is desired. Two variants are currently implemented:

- "Points" type is

  point-many-unordered-rigid-discrete
- "TimestampedPoints" type is

  point-many-chronological-rigid-discrete

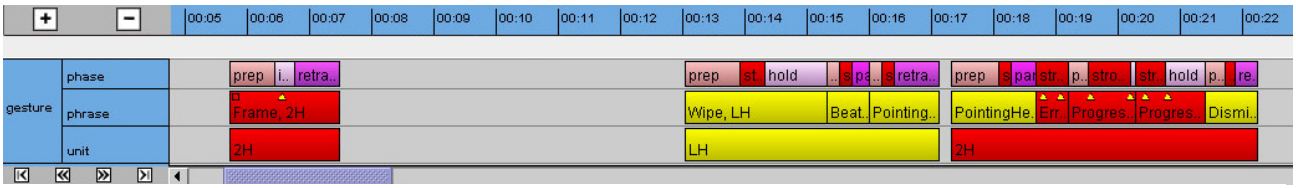


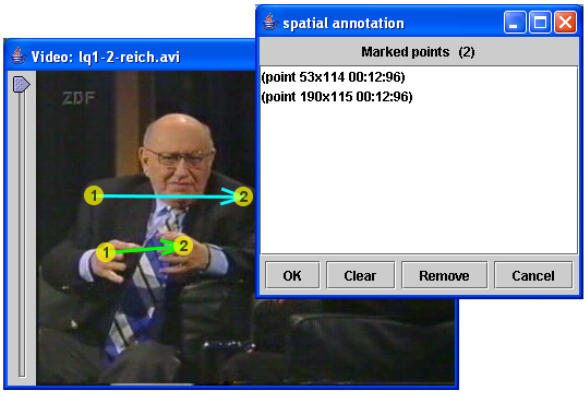
Figure 5: The annotation board during spatial coding remains active to allow the user to select the timestamps, visualized as small yellow triangles (middle track).

In order to use this feature the user has to add a spatial attribute in the *coding scheme* (called *specification* in ANVIL):

```
<track-spec name="phrase" … >
    …
    <attribute name="2H-distance"
               valuetype="TimestampedPoints" />
    …
</track-spec>
```

In ANVIL's graphical interface, when the user wants to edit such an attribute a window pops up that displays any annotated points in a list (Figure 6). To add points the user can move on the video screen and double-click to add a point (a single click positions the crosshair). At the same time, the annotation board remains "active" so that the user can specify the exact time point (in case of timestamped points, see Figure 5). This gives the user two handles: on time (annotation board) and space (video screen).

Figure 6: The user can draw directly on the video screen. Marked points are immediately visualized as yellow (numbered) dots and are also listed in a separate window



in a list.

As opposed to a 3D interface (Figure 4), the two ANVIL views can be considered two orthogonal views on the three dimensional timeline-screen space depicted in Figure 4.

6. Conclusion and Future Work

We introduced spatiotemporal coding of video material in the ANVIL tool. It allows to add a spatial refinement to the annotation by drawing directly on the video screen. A taxonomy of possible variations in spatiotemporal annotation was presented. We argued that the vidoe plus timeline view gives a good and intuitive handle on the complexity added by spatiotemporal coding. Currently, the ANVIL tool allows the coding of timestamped points with optional linear interpolation between points. This feature has already been used coding human gestures (Kipp et al., 2008) and has potential for many other research areas, e.g. semantic coding for video-based information retrieval. Extending this feature to other shapes and interpolation types will be the subject of future work as well as export to the MPEG-7 format.

7. Acknowledgements

This research has been carried out within the framework of the Excellence Cluster *Multimodal Computing and Interaction* (MMCI), sponsored by the German Research Foundation (DFG).

8. References

Bigbee, A., Loehr, D., and Harper, L. (2001) Emerging Requirements for Multi-Modal Annotation and Analysis Tools. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*.

Kipp, M., Neff, M. und Albrecht, I. (2008). An Annotation Scheme for Conversational Gestures: How to economically capture timing and form. In: *Multimodal Corpora for Modelling Human Multimodal Behavior*, Special issue of the International Journal of Language Resources and Evaluation. Springer.

Kipp, M. (2004). *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation*. Boca Raton, Florida: Dissertation.com.

Kipp, M. (2001). Anvil - A Generic Annotation Tool for Multimodal Dialogue. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1367-1370.

Martin, J.C., Kipp, M. (2002). Annotating and Measuring Multimodal Behaviour - Tycoon Metrics in the Anvil Tool. In: *Proceedings of the Third International Conference on Language Resources and Evaluation* (LREC).

Martinez, J.M., Koenen, R., Rereira, F. (2002). MPEG-7: the generic multimedia content description standard, part 1. In: *IEEE Multimedia* 9 (2), pp. 78-87.

Rohlfing, K., Loehr, D., Duncan, S., Brown, A., Franklin, A., Kimbara, I., Milde, J.-T., Parrill, F., Rose, T., Schmidt, T., Sloetjes, H., Thies, A., Wellinghoff, S. (2006). Comparison of multimodal annotation tools: workshop report. In: *Gesprächsforschung, Online-Zeitschrift zur verbalen Interaktion* (7), pp. 99-123.