BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025

http://excel.fit.vutbr.cz

SYNCHRONIZATION OF SLIDES WITH A VIDEO RECORD-ING OF A LECTURE

Dan Valníček

Abstract

This paper aims to make a program for automatic slide-to-video interval annotations on video lecture recordings. This is useful when a student doesn't understand the problem shown on the slide just by looking at the slide, and needs to hear the lecturer's explanation. The program uses the Scale Invariant Feature Transform (SIFT) [1], RANdom SAmple Consensus (RANSAC) [2] homography estimation, and Cosine Similarity [3] to match slides to frames in the video. A great feature is the direct insertion of intervals matched to the PDF, which was made by creating a custom PDF extension. The slide-matching algorithm was tested on 41 385 frames with 96 % passability, making the application very usable though not perfect. Lastly, a custom PDF viewer and video player application was made and released, allowing students to try it out. The application is free to download from GitHub and can be used by anyone.

*xvalni00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. 1. Automatic slide annotation

Fig. 1 shows the slide-frame annotation process in a black-box manner. The input of this algorithm consists of the slides in Portable Document Format (PDF) and a video recording of a lecture with the same slides. The output consists of PDF with video intervals that have been found to contain a given slide being presented.

2. 2. Slide-frame matching

The slides are matched to frames in three main steps:

2.1 Feature matching

Fig. 2 Features are detected and described using the SIFT descriptors. Every descriptor is matched to the nearest descriptor in the database of slide descriptors. The matches are pruned by the Lowe's distance ratio test (section 7.1 [1]). The matches left after pruning are attributed to their respective slides.

2.2 Homography transformation and verification

Homography, a perspective matrix mapping the points in the slide to points in the frame, is estimated using RANSAC. The quality of mapping is further evaluated by testing if transformed boundaries of the frame fully overlap with boundaries of the original slide, which was not transformed.

Orange rectangle – represents the edges of the slide being compared in the PDF file and is never transformed.

Light blue quadrilateral – represents edges of the frame from video.

Fig. 3a shows the boundary verification result of the homography made with the correct slide, passing the test.

Fig. 3b shows the boundary verification result of the homography made with the incorrect slide, failing the test.

2.3 Second feature matching and Cosine Similarity evaluation

The SIFT descriptors are detected on the transformed image pruned by RANSAC again, and the ones left are added to a list of matched unique descriptors (ul).

In the end, this list is taken and Cosine Similarity is calculated with all slides by this formula:

$$sim = \frac{|m_{sl}|}{\sqrt{|d_{ul}|} \cdot \sqrt{|d_{sl}|}} \qquad (1)$$

- $|m_{sl}|$ – the number of matches in the list of unique descriptors with a given slide.

- $|d_{ul}|$ – length of the list with unique matched descriptors across all slides.
- $|d_{sl}|$ – number of all descriptors existing for a given slide.

The slide with the highest similarity is attributed to the current timestamp, unless the similarity is under a threshold, leaving the timestamp with no slide attributed.

Fig. 4a shows the number of descriptors successfully matched with the correctly matching slide in Fig. 4b. Almost all descriptors were found to match the slide descriptors.

3. Test evaluation

The testing was done against a self-made dataset from BUT FIT lecture recordings and slides provided by the lecturers. The dataset included five lectures from different classes across 3 lecture halls, totaling 41 385 frames.

Chart 1 shows slide matching the tested precision against the dataset at different slide-matching algorithm versions.

3.1 Version 1

The first version would only transform the image using homography (section 2.1) and the sum of point weights, given by the distance between the correct coordinates and the transformed point's coordinates. The final score for the given slide was given by normalizing the sum mentioned in the previous sentence by the number of descriptors for the given slide.

3.2 Version 2

The second version added the second SIFT detection after perspective transformation, boundary verification step (sections 2.2, 2.3), and used Term Frequency-Inverse Document Frequency (TF-IDF) [4] weighting for matched descriptors. The weighting by TF-IDF was proven to be detrimental to the results.

3.3 Version 3

The latest version removed the TF-IDF weighting and used descriptor counts in the similarity calculation explained in section 2.3.

4. Viewer application

The application was made with the student use case in mind. A custom PDF developer extension was made and registered for this application, which enables writing of the intervals right into the PDF file. The application allows viewing any PDF files, but if the PDF file contains a custom extension, it will show slide intervals for the video lecture where the slides appear.

Fig. 5 The application combines a video player with a PDF viewer. The main feature is the ability to see a list of intervals where the slide appears and click to skip the video to that section.

Other features include:

- having a video annotated and exported as a PDF with annotations,
- fast scrolling to the slide currently shown in the video,
- and automatic PDF scrolling with the video.

The application was released on GitHub[1] and can be freely downloaded for Windows and compiled for Linux.

5. Conclusions

In conclusion I think that this is a good project with good idea, but there are some things that I would like to improve. First of all, I would like to improve the slide matching accuracy, the current 96 % is pretty good but I believe it can be even better. To achieve this, I would study other descriptor comparison methods.

Another area that needs improvement is the matching speed, because right now it takes around 0.5 seconds to match a single frame, meaning that at a 1 Hz sampling rate, it takes about half the video length to create all annotations. To improve matching speed, a study on different feature detectors and their performance should be made to find a feature detector with better speed. Another possible optimization could be a rewrite from Python to C++.

The PDF extension could be improved to contain more data, like a hash of the video, download URL, and maybe some student notes specific to the video.

The last area that needs improvement is the PDF viewer and video player user experience and maybe user interface.

References

[1] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[1] https://github.com/DanValnicek/slide-lecture-sync

[2] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[3] Wikipedia. Cosine similarity — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Cosine%20similarity&oldid=1283628649, 2025. [Online; accessed 12-April-2025].

[4] Wikipedia. Tf–idf — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Tf%E2%80%93idf&oldid=1268530811, 2025. [Online; accessed 13-April-2025].