AMIGO – Automatic Indexing of Lecture Footage

Markus Eberts
RheinMain University of Applied Sciences
Email:markus.b.eberts@student.hs-rm.de

Adrian Ulges
RheinMain University of Applied Sciences
Email:adrian.ulges@hs-rm.de

Ulrich Schwanecke
RheinMain University of Applied Sciences
Email:ulrich.schwanecke@hs-rm.de

***Abstract*—We present AMIGO, an automatic indexer for video presentations which – given an e-lecture and supplementary slides – localizes the exact time and position of each slide displayed in the video footage. This offers richer access to viewers, including a slide-accurate navigation and a text-based interaction with the video. AMIGO is based on a matching of local features between video frames and presentation slides. Our key contribution, however, is the combination of local feature matching with two temporal models (a Hidden Markov Model (HMM) and a simple heuristic filter), exploiting the alignment of the presentation with the reading order of its supplementary material. We demonstrate the effectiveness of our approach in quantitative experiments on a dataset of e-lectures and screencasts, which show – with an average accuracy of over 95% – that the approach works under occlusion and camera motion.**

I. Introduction

E-learning has become a major trend over the recent years, and conventional knowledge transfer is increasingly enhanced or replaced by electronically supported forms of teaching. Learners – nowadays digital natives – are open to engaging with new media, and can benefit from a more flexible time management and learning speed, as well as from an individualized learning process. Thereby, a key-driver of e-learning are educational videos. These range from lecture recordings over screencasts to professionally produced webcasts, and constitute the core of most online courses[1]. A key problem with video, however, lies in the fact that it offers only limited interaction possibilities, and cannot easily be linked with additional material such as presentation slides, scripts, notes, forum posts and discussions. Learning as an interactive process, however, requires massive interaction, such as navigation (*where in the video does Section 3 start?*), fine-grain access to certain pieces of information (*where can I find the example for Newton's method?*), storage and reorganisation (*can I copy this piece of text from the video?*), or exploration (*where can I find additional material?*).

Up to some extent, video can be made more accessible by a manual annotation and/or subtitling, or by a fine-grain segmentation into short subsegments of 1-2 minutes. This is, however, extremely time-consuming and involves considerable costs. Therefore, this paper addresses the automatic indexing of educational videos as an alternative. Our approach covers both regular screencasts and the more challenging case of lecture recordings, even with a moving camera. It is based on a local feature matching between the video stream and the documents visible in it (typically, presentation slides), which localizes the exact time and position of each slide/page in the video. This offers a rich set of interactions, ranging from slide-accurate navigation over text search in the video – or copy-and-paste from the video – to the recommendation of add-on content.

Our key contributions are: First, we propose an automatic indexing for lecture slides based on local feature matching, an approach which has not been followed before to the best of our knowledge. Second, our model includes a novel combination of feature-matching with temporal models, enforcing the alignment of the video content with the reading order of the supplementary material. Third, we demonstrate in quantitative experiments that our approach is effective for webcasts as well as lecture recordings, with an average matching accuracy of 95.05%. Errors are caused by corner cases such as multiple slides visible at once, or strong occlusion. A web demo of the system is available online[2].

II. Related Work

Interactive online courses are increasingly popular, offered by institutions of higher education as well as dedicated web-based portals (e.g. the Kahn Academy, Udacity, Coursera, or video2brain). In practice, interaction with e-lectures is usually limited to standard features like *play* or *fast forward*. Besides the video footage, additional exercises, slides, or quizzes are offered. Indexing of video content is usually done by a segmentation into short subsequences, to which headlines, tags, and subtitles are (manually) added [1].

Automatic indexing of e-lectures is conducted in research prototypes [2], [3] and opencast systems [4]. The footage is usually required to show a rectified, high-quality version of the slides, as it can be captured by opencast systems recording the VGA stream with specialized hardware during the presentation. In this case, the VGA stream is aligned with the audio stream, optical character recognition (OCR) is applied, and slide transitions are detected, which allows a frame-accurate text search. In contrast to this, our approach is based on a local feature matching [5] between video and documents, which offers several benefits: First, no additional hardware is required while recording the presentation. Second, our approach is independent of text and can also match graphic content or imagery. Third, OCR is replaced with a more robust localization of the whole slide, which works even in case of perspective distortions, difficult lighting conditions, partial occlusion, inconventional slide designs (using exotic fonts or small text sizes) as well as poor video quality and resolution. Text can be extracted directly from the slides (using, e.g., pdf libraries) free from misrecognition.

[1] See e.g. `udacity.com`, `inversity.org`, `coursera.org`, or `khanacademy.org`

[2] `http://bolg.cs.hs-rm.de:8000/videoPlayer`

From a computer vision point of view, our work is based on local feature matching, stable homography estimation by RANSAC like approaches as [6], and geometric constraints [7]. Thereby, we address an object recognition problem, which is simplified by the fact that the target objects (the slides) are planar and (usually) well-textured. Local feature matching has been successfully used for recognition of planar objects[3] and document matching [8]. Beyond this, we integrate the noisy results with an HMM-based model that combines the match scores with temporal constraints, enforcing the alignment of the recognized slides with the expected reading order.

III. Approach

We subsample a given e-lecture (e.g., at 1 fps), obtaining a sequence of video frames $\mathcal{F}$ to be indexed. Supplementary documents (typically presentation slides) are assumed to be given in pdf format. Potentially, there are multiple documents accompanying a video. We collect all pages (or *slides*) from all documents in a set $\mathcal{S} = \{s_1, ..., s_n\}$. To *index* the e-lecture means to infer a mapping from $\mathcal{F}$ to $\mathcal{S} \cup \{s_0\}$, whereas s_0 indicates that no slide is visible in a given frame.

A. Keypoint Extraction, Matching, and Score Computation

Each frame to be indexed as well as the slides from the supplementary material are rendered into images[4]. Next we extract local features using SIFT [9] from each frame and slide, and conduct a local feature matching for each slide/frame combination by finding the most similar key point in the frame to each one in the slide based on an approximate nearest neighbor search[5] [10] . As a result, we obtain a set of local correspondences (or *matches*). We perform filtering and reasoning on these matches to infer the correct slide for each frame. First, as the original set of matches must be expected to contain a substantial amount of false positives, we apply several filters to improve match quality:

1) We apply a ratio test [9] based on the key points' descriptors, i.e. we filter matches with $d_1/d_2 > \gamma \in]0, 1[$, where d_1, d_2 are the descriptor distances of the nearest and the second nearest neighbors respectively.
2) We estimate the dominant homography [7] $\mathcal{H}$ mapping key point positions between the slide and the frame using RANSAC [11]. Any matches whose positions do not match $\mathcal{H}$ are discarded.
3) We validate $\mathcal{H}$ using simple geometric criteria. First, since slides should not be flipped, $\mathcal{H} = (h_{ij})$ should be orientation-preserving, i.e. $\det \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} > 0$ [12]. Otherwise we discard *all* matches.
4) Since slides should appear at prominent size, we map the slide s into the frame using $\mathcal{H}$ and compare the resulting area $|\mathcal{H}(s)|$ with the frame's total area $|f|$. If $|\mathcal{H}(s)| < \beta \cdot |f|$, we discard all matches.
5) As slides should roughly be of rectangular shape in the video, we expect all internal angles of the mapped slide $\mathcal{H}(s)$ to be close to $90°$ and discard all matches containing an inner angle α with $|\alpha - 90°| > \delta$.

[3]See `www.ltutech.com/solutions/mobile-visual-search/`
[4]Frames are rendered by OpenCV, slides by Wand (`docs.wand-py.org`)
[5]Our implementation uses FLANN (`www.cs.ubc.ca/research/flann`)

Based on the set of refined matches between frame f and slide s, we estimate a *score* indicating the likelihood that f does in fact display s. Intuitively, the higher the number of matches, the higher this score, i.e. $score^{\text{matches}}(s, f) := n(s, f)$, where $n(s, f)$ denotes the number of matches after refinement. This absolute count of matches, however, may not be the best option: First, for strongly textured slides/frames, it is intrinsically higher due to false positives. Second, as scores will serve as input features for later reasoning steps, we expect their distribution to be consistent across a wide range of videos, i.e. scores should be robust to illumination, texturing, or occlusion. As we suspect (and will validate later), this second criterion is not met by the absolute match count. Therefore, we suggest an additional *normalization* with the number of keypoints per slide $n(s)$ and the number of keypoints per frame $n(f)$, yielding

$$score^{\text{norm}}(s, f) := \frac{n(s, f)}{n(s)} + \frac{n(s, f)}{n(f)}. \quad (1)$$

B. Hidden Markov Model

Given the score measures from Section III-A, a simple indexing strategy might be to pick the slide with maximum score for each frame (or no slide if all scores are below a certain threshold). However, we still must expect errors when following this strategy, due to clutter, redundant slide content, or occlusion. To infer the most plausible slides for each frame, automatic indexing should exploit both the image match quality *and* the *order* of the slides in the supplementary material. Thereby, a probabilistic approach seems suitable to handle uncertainty: Sometimes the presenter may violate the reading order (e.g. skipping slides to answer a question), and sometimes the match quality between slide and frame may be poor (for example in case of partial occlusion).

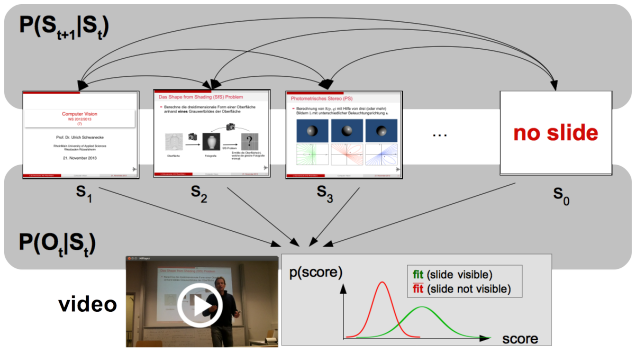

Fig. 1. In our approach, HMM states correspond to the presentation slides. Transition probabilities model the likelihood of switching slides, and output probabilites are derived from the match quality between video frame and slide.

To address these challenges, we suggest HMMs, a well-known approach towards the recognition of sequential data with many applications in Computer Vision [13]. HMMs assume input data to come as a sequence of *observations* $o_0, o_1, o_2, ...$ and infer a corresponding sequence of *states* (modeled as random variables $S_0, S_1, S_2, ...$). Inference is based on the three probability distributions

- $P(S_0 = s)$, the distribution of the initial state,
- $P(S_{t+1} = s_{t+1}|S_t = s_t)$, the transition probability between states, and

- $P(o_t|S_t = s)$, the *output* probabilities of observing certain feature values, given a state.

In our case, each time step t corresponds to a new video frame to be indexed, the corresponding state S_t is the *slide* to estimate, and the observations o_t will be derived from the scores outlined in Section III-A. Figure 1 illustrates this approach. Formally, we define a set of states $\mathcal{S}' := \mathcal{S} \cup \{s_0\}$ (where $\mathcal{S}$ contains all slides from all documents, and s_0 represents situations in which no slide is visible). To model the HMM's output probabilities, we distinguish two cases: fit (a slide s is visible) and $\overline{fit}$ (the slide s is not visible). We assume annotated lectures for a supervised training to be given, from which we can derive frame-slide pairs for both classes. We choose one of the score computation measures from Section III-A and derive a set of training scores, from which we estimate the means $\mu_{fit}, \mu_{\overline{fit}}$ and variances $\sigma^2_{fit}, \sigma^2_{\overline{fit}}$ for both classes using maximum likelihood estimation. Note that we expect μ_{fit} to be substantially higher than $\mu_{\overline{fit}}$, as the number of matches between slide and frame should increase in case the slide is actually visible. To model the observation vector at time t, we match the frame f_t with all slides and collect the resulting scores, i.e.

$$o_t := \Big(score(s_1, f_t), score(s_2, f_t), ..., score(s_n, f_t)\Big).$$

We choose a multivariate normal distribution for o_t, assuming s_i is visible, i.e. $p(o_t|S_t = s_i) := \mathcal{N}(o_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with the parameters

$$\boldsymbol{\mu}_i := \Big(\mu_{\overline{fit}}, ..., \mu_{\overline{fit}}, \underbrace{\mu_{fit}}_{\boldsymbol{\mu}_i^i}, \mu_{\overline{fit}}, ..., \mu_{\overline{fit}}\Big)^T,$$

$$\boldsymbol{\Sigma}_i := Diag\Big(\sigma^2_{\overline{fit}}, ..., \sigma^2_{\overline{fit}}, \underbrace{\sigma^2_{fit}}_{\boldsymbol{\Sigma}_i^{ii}}, \sigma^2_{\overline{fit}}, ..., \sigma^2_{\overline{fit}}\Big).$$

Corresponding to this distribution, when in state i, we expect the scores for all slides to be low except for slide i. In case of s_0, we choose *all* parameters from the $\overline{fit}$ distribution, i.e.

$$\boldsymbol{\mu}_0 := \Big(\mu_{\overline{fit}}, ..., \mu_{\overline{fit}}\Big)^T, \quad \boldsymbol{\Sigma}_0 := Diag\Big(\sigma^2_{\overline{fit}}, ..., \sigma^2_{\overline{fit}}\Big).$$

We refrain from learning the HMM's initialization and transition probabilities, as we expect them to vary between presentations (e.g. depending on the teacher's speed or questions from the audience). Instead, we use a simple strategy based on manually chosen parameters: Given a small probability $\epsilon > 0$ to model "unlikely" events, we choose the initialization distribution as

$$P(S_0 = s) \propto \begin{cases} p_s & \text{if } s \text{ is the first slide of its document} \\ 1 - p_s & \text{if } s = s_0 \text{ (no slide visible)} \\ \epsilon & \text{else.} \end{cases}$$

To normalize, we divide the above values by their total. Furthermore, to account for multiple documents, we divide all probabilities by the number of documents, i.e. we favor no certain order of documents and do not consider their length.

To model the HMM's transition probabilities, we assume all slides to be watched sequentially and for an equal duration, which leads to

$$p := \frac{n}{T}$$

as the probability for switching from any slide s to its successor. Thereby, T is the number of frames to index in the video and n denotes the overall number of slides. Based on p, we choose the following transition probabilities:

$$P(S_{t+1} = s_{t+1}|S_t = s_t) \propto \begin{cases} p & \text{if } s_{t+1} \text{ is } s_t\text{'s successor} \\ \lambda \cdot p & \text{if } s_{t+1} \text{ is } s_t\text{'s predecessor} \\ \nu \cdot p & \text{if } s_{t+1} = s_0 \text{ (no slide)} \\ 1 - (1 + \lambda + \nu) \cdot p & \text{if } s_{t+1} = s_t \text{ (stay on slide)} \\ \epsilon & \text{else (any other slide in any document)} \end{cases}.$$

assuming that $s_t \neq s_0$, i.e. some slide was visible before the transition. In case *no* slide was visible, we expect any subsequent slide to be equally likely, resulting in

$$P(S_{t+1} = s|S_t = s_0) \propto \begin{cases} 0.8 & \text{if } s = s_0 \\ \frac{0.2}{n} & \text{else.} \end{cases}$$

Again, we normalize the above values to sum up to one, so we obtain a proper probability distribution.

C. State Filtering

Finally, we apply an additional postprocessing step based on a heuristic filtering of states. To be of interest to the video's viewers, a slide should be visible for at least a few seconds. Therefore, we remove short subperiods (e.g., when quickly skipping through slides): Starting at the beginning of the video, we check all its *segments*, i.e. all subperiods in which a certain slide is visible without interruption. We sort these segments by their length and start with the shortest one. Whenever the duration of a segment is smaller than τ seconds, we merge the segment with its predecessor, i.e. the slide recognized by AMIGO is replaced with the slide that was last visible before. Effectively, this heuristic filtering removes short subperiods.

IV. Experiments

We evaluate our automatic indexing on a set of e-lectures and screencasts recorded by the authors at RheinMain University of Applied Sciences. An overview of the data is given in Table I. The dataset contains 203 video minutes in total from the lecture "Shape from Shading" (SfS) from the course "Computer Vision" (CV) and five different lectures from the course "Analysis". The courses were held in German, recorded in different auditoriums, show different slide designs, and were filmed with different cameras and from different viewpoints. The "Computer Vision" video was recorded with a fixed camera at resolution 840×480 px and displays substantial occlusion by the lecturer. The "Analysis" videos are of resolution 1280×720 px, include camera motion, and contain several minutes of desktop camera recordings, which display close-ups of the presenter writing onto the slides (see Figure 2 for examples). In case a video includes manual writing,

Course	Topic	video type	length (min)	#pages	#visible
CV	SfS	e-lecture	101	42	42
Analysis	Motivation	e-lecture	10	33	6
Analysis	Newton	e-lecture	22	33	7
Analysis	Bisection	e-lecture	21	33	10
Analysis	Regula Falsi	e-lecture	23	31	11
Analysis	Taylor series	screencast	26	13	13

TABLE I. VIDEOS USED IN THE EXPERIMENTS (203 MINUTES TOTAL).

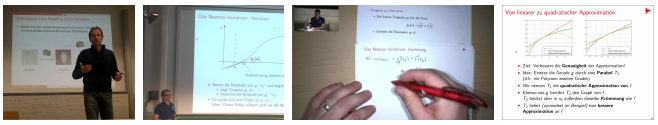

Fig. 2. From left to right: Sample frames from our test videos, showing the e-lectures "Computer Vision" , "Analysis", and a screencast.

its slides were scanned after the lecture, and the scans were used for indexing. Otherwise, the original pdfs rendered with LaTeX were used. Finally, as a proof of concept, we also include a screencast produced for the "Analysis" course.

Accuracy Measures Indexing was conducted at 1 fps, amounting for 12,164 frame-slide pairs to estimate in total, which were annotated manually to assess the accuracy of indexing. We use two different quality indicators: First, we measure the percentage of frames for which the correct slide (or the correct "no-slide" state s_0) was recognized. We refer to this measure as **state accuracy** in the following. Second, we measure the correctness of state *transitions*. Thereby, a *transition* occurs whenever the slide displayed in the video changes. Given the set of true transitions $\mathcal{T}$ in the video and the set recognized by AMIGO $\mathcal{T}'$, we measure the **Jaccard index** $J(\mathcal{T},\mathcal{T}') := {}^{(\mathcal{T}\cap\mathcal{T}')}/_{(\mathcal{T}\cup\mathcal{T}')}$. Thereby, two transitions are assumed equal if the switch recognized by AMIGO does not deviate by more than 3 seconds from the true one.

Parameters We set the angle tolerance to $\delta = 10°$, the area tolerance β (Subsection III-A) to 10%, and the ratio test parameter γ (Subsection III-A) to 0.75. Any parameters regarding SIFT extraction and representation were set to their OpenCV default values. The slides to be recognized were rescaled to fit the video resolution before matching. Matching itself was conducted using FLANN. We set the HMM's parameter to model small probabilities to $\epsilon = 10^{-16}$. The start probability of a document's first slide was chosen as $p_s = 0.9$, the transition probabilities were set to $\lambda = {}^1/_8$ (for going back to the preceeding slide) and to $\nu = {}^1/_{15}$ (for switching to the "no-slide" state). State filtering used a value of $\tau = 10$ seconds.

A. Score Measures

Figure 3 illustrates the distribution of scores of frame-slide pairs from two different lectures, using the absolute match count $score^{\text{matches}}(s,f)$ (see Section III-A) as opposed to the normalized scores $score^{\text{norm}}(s,f)$ (see Equation (1)). For each video, we collected fit pairs where the slide is visible in the frame, computed the resulting match scores, and estimated the mean and variance of the scores' distributions. The resulting normal distributions are illustrated in the plots. When using absolut match counts the distributions can differ significantly between the videos. Here, the scores cannot be considered robust when training on different videos than testing on, and the HMM cannot be expected to generalize well. However, when using keypoint normalization instead, the corresponding distributions are a lot more similar. Therefore, for the normalized match count a higher robustness can be expected than for the absolute match counts. For that reason, we use the normalized scores in the following experiments.

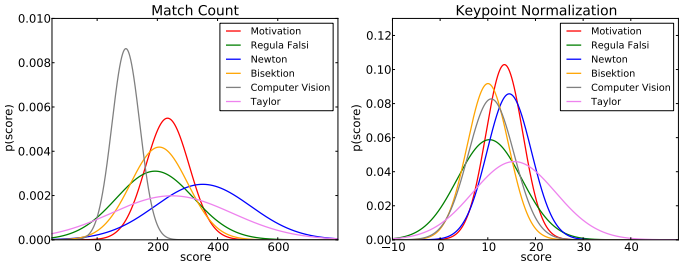

Fig. 3. The distribution of "fit" scores in the different test videos, using $score^{\text{matches}}(s,f)$ (left) and $score^{\text{norm}}(s,f)$ (right).

B. Recognition Results

Overall quantitative results of indexing are illustrated in Table II. We tested three different methods (all using $score^{\text{norm}}$):

1) **Baseline**: A plain feature matching, including a ratio test and a filtering using a homography, but without any additional homography-based validation (only points 1 and 2 from the refinements in Section III-A).
2) **Homography Validation**: In addition to 1., we include the homography-based checks for orientation, area and angle. If a frame-slide pair fails any criterion, all its matches are discarded (effectively setting its score to 0).
3) **Homography Validation & HMM**: In addition to the homography validation, we refine the resulting slide estimates using the HMM approach from Section III-B. The HMM was trained on the "Computer Vision" video (we will address generalization of the model between different videos later).
4) **Final**: Same as the last run, but now including the final *state filtering* outlined in Section III-C.

Table II shows that a plain baseline matching is inaccurate: Mostly it produces incorrect slide transitions (the Jaccard index is 3.82% on average) and recognizes the correct slide only in 73% of cases. Homography checks very effectively improve the results (23.20% / 92.05%). Beyond this, our results also demonstrate that accuracy can be improved further by including time contraints: The HMM model results in an average Jaccard index of 50.76% and state accuracy of 93.45%. Finally, state filtering improves results further to 76.72% / 95.05%. In particular, the *combination* of both temporal models leads to the best results, outperforming both HMM (Table II) and heuristic state filtering alone (71.80% / 94.35%).

Figure 4 illustrates the effects of the temporal models on a subsection of the *Regula Falsi* video. The x-axis indicates time (seconds), the y-axis the visible/recognized slide. Gray segments indicate the ground truth, red ones recognition results. For green segments, both are well aligned, i.e. recognition is correct. We observe that – when using no additional time constraints – recognition tends to be unstable, leading to lots of fragmented subperiods with many false slide transitions (which explains the low Jaccard index, in particular). Using the HMM and state filtering, these errors are smoothed out. The time-based models, however, also produce a longer incorrect segment in the beginning. The reason is redundant content, as illustrated in Figure 5, i.e. multiple slides are visible at once.

Course	Topic	baseline		homography validation		homography validation & HMM		final	
		jaccard index	state accuracy	jaccard index	state accuracy	jaccard index	state accuracy	jaccard index	state accuracy
CV	SfS	1.94	59.58	28.18	96.75	73.75	98.24	93.02	98.96
Analysis	Bisection	2.65	66.45	26.32	91.67	45.45	92.39	64.71	96.31
Analysis	Newton	4.81	71.60	16.07	93.45	45.00	95.39	60.00	96.96
Analysis	Motivation	4.35	76.97	33.33	95.76	77.78	97.37	100.00	99.18
Analysis	Regula Falsi	3.30	75.16	26.98	85.86	47.37	86.32	69.23	87.74
Analysis	Taylor series	5.89	88.33	8.33	88.85	15.19	90.99	73.33	91.18
average		**3.82**	**73.02**	**23.20**	**92.05**	**50.76**	**93.45**	**76.72**	**95.05**

TABLE II. RESULTS (LEFT TO RIGHT): RAW MATCHING, HOMOGRAPHY-BASED VALIDATION, ADDING THE HMM, AND ADDITIONAL STATE FILTERING.

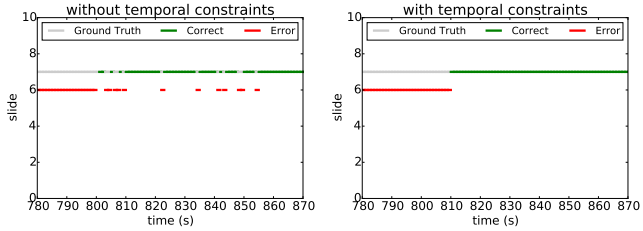

Fig. 4. Left: Plain frame-wise matching leads to unstable recognitions. Right: Results are stabilized by the temporal constraints of HMM and state filtering.

C. Generalization

AMIGO's automatic indexing must generalize under a wide range of recording conditions, recording hardware and slide designs. These parameters might affect the HMM's observation densities, whose means and variances are learned from manually annotated sample videos. To validate generalization between different recording conditions, we test AMIGO (using the *final* setup, i.e. including the HMM and state filtering) on videos from the two courses "Computer Vision" and "Analysis", and evaluate indexing accuracy when training the HMM on the *same* course as the video belongs to, and on the *other* course. In the latter case, state accuracy drops slightly (from 95.28% to 95.05%), while the Jaccard index (which allows a tolerance of 3 seconds) is not affected at all.

D. Error Inspection

In an in-depth inspection of error cases we found 8 video subsequences for which slide recognition remained error-prone. Figure 5 illustrates the sources of error discovered: **Partial occlusion**: Manually, "no slide" was annotated, but AMIGO recognizes the partially occluded slide. **Redundant content**: Multiple slides are visible simultaneously. AMIGO picks the slide at the top, while the slide at the bottom is actually in the limelight. **Lack of texture**: Animated slides or slides written on manually may show little texture in the beginning. Sometimes, no slide can be found in these cases. **Missing content**: A slide is visible that was not provided with the course material. A similar (but incorrect) slide from the official course material is picked.

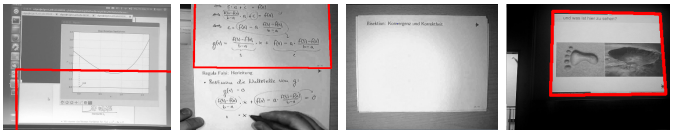

Fig. 5. Error cases occur due to (from left to right): partial occlusion, redundant content, lack of texture, or missing content.

V. CONCLUSION

We have presented AMIGO, an automatic indexer for educational video content, which localizes the exact time and position at which presentation slides occur in video footage, even under partial occlusion and camera motion. Our approach is based on an image matching technique using local SIFT features. It extends over a plain frame-wise matching by temporal constraints, using an HMM and heuristic state filtering, which is shown to improve the accuracy of recognition. We also demonstrate the robustness of indexing on a set of e-lectures and screencasts: Few errors were found, which were all caused by corner cases such as occlusion or redundant content.

Next we will focus on exploiting indexing results for user interaction. We plan to implement text-based interaction (including text search, copy-paste, or filling in text that is occluded in the video footage). Other features may include the exploitation of speech recognition for indexing, or even more fine-grain access (e.g. clicking on a point on a slide, and directly navigating to the exact second where it is explained). Finally, to eliminate the remaining misrecognitions, a user interface for inspecting and refining indexing results is required.

REFERENCES

[1] L. R. et al., "BIBS: A Lecture Webcasting System," Berkeley Multimedia Research Center, Tech. Rep., 2001.

[2] J. Adcock, M. Cooper, L. Denoue, H. Pirsiavash, and L. A. Rowe, "Talkminer: A lecture webcast search engine," in *Proc. ACM Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 241–250.

[3] H. Yang, C. Oehlke, and C. Meinel, "An automated analysis and indexing framework for lecture video portal," in *Proc. ICWL*, 2012.

[4] "Matterhorn: Open Source Lecture Capture and Video Management for Education," opencast.org/matterhorn (retrieved: August'14).

[5] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. Trends. Comput. Graph. Vis.*, pp. 177–280, Jul. 2008.

[6] S. Otte, U. Schwanecke, and A. Zell, "ANTSAC: A Generic RANSAC Variant Using Principles of Ant Colony Algorithms," in *ICPR*, 2014.

[7] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2005.

[8] M. Iwamura, T. Nakai, and K. Kise, "Improvement of retrieval speed and required amount of memory for geometric hashing by combining local invariants," in *BMVC2007*, vol. 2, Sep. 2007, pp. 1010–1019.

[9] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[10] M. Muja and D. Lowe, "Scalable Nearest Neighbor Algorithms for High Dimensional Data," *TPAMI*, vol. 36, pp. 2227–2240, 2014.

[11] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[13] H. Bunke and T. Caelli, Eds., *Hidden Markov Models: Applications in Computer Vision*. River Edge, NJ, USA: Wspc, 2002.