A System that Learns to Tag Videos by Watching Youtube

Adrian Ulges[1,2], Christian Schulze[2], Daniel Keysers[2], Thomas M. Breuel[1,2]

[1] Department of Computer Science, Technical University of Kaiserslautern
{a_ulges,tmb}@informatik.uni-kl.de
[2] Image Understanding and Pattern Recognition Group
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern
{schulze,keysers}@iupr.dfki.de

Abstract. We present a system that automatically tags videos, i.e. detects high-level semantic concepts like objects or actions in them. To do so, our system does not rely on datasets manually annotated for research purposes. Instead, we propose to use videos from online portals like `youtube.com` as a novel source of training data, whereas tags provided by users during upload serve as ground truth annotations. This allows our system to learn autonomously by automatically downloading its training set.
The key contribution of this work is a number of large-scale quantitative experiments on real-world online videos, in which we investigate the influence of the individual system components, and how well our tagger generalizes to novel content. Our key results are: (1) Fair tagging results can be obtained by a late fusion of several kinds of visual features. (2) Using more than one keyframe per shot is helpful. (3) To generalize to different video content (e.g., another video portal), the system can be adapted by expanding its training set.

keywords: content-based video retrieval, automatic video annotation, online videos

1 Introduction

During the last years, online video has evolved as a source of information and entertainment for users world-wide. For an efficient access to this video data, most commercial providers rely on text-based search via user-generated tags – an indexing that requires manual work and is thus time-consuming and incomplete.

In parallel, content-based techniques have been developed that try to use the content of a video to infer its semantics. To achieve such "tagging", i.e. an automatic annotation of videos with high-level concepts like objects, locations, or actions, systems are usually trained on a set of labeled videos. Acquiring such ground truth information manually is costly and poses a key limitation for the practical use of automatic annotation systems.

In this paper, we introduce a system that learns to tag videos from a different kind of training data, namely by watching video clips downloaded from online video portals like `youtube.com`. Thereby, tags provided by users when uploading content serve as ground truth annotations. Our work is thus targeted at online video (1) as an *application* (our system proposes adequate tags for videos and can thus support users with tagging or keyword search), and (2) as a *data source* for visual learning: online videos are publicly available and come in a quantity that is unmatched by datasets annotated for research purposes. We envision this data to complement (or even replace) existing training sets.

Despite the enormous diversity of online video content and the fact that it shows lots of irrelevant scenes (as is illustrated in Figure 1), we present a prototype that shows how visual learning of semantic concepts from online video is possible (a demo can be found at http://demo.iupr.org/videotagging). Compared to our previous workshop publication [17], we present several novel quantitative experiments with our prototype on large-scale datasets of real-world online videos. Our key results are the following: (1) A fusion of multiple feature modalities is essential for a successful tagging. (2) Using more than a single keyframe per shot improves tagging performance. (3) The system adapts well to different video data if expanding its training set.

2 Related Work

An area strongly related to our work via the use of keyframes is automatic image annotation, which has been dealt with by modeling latent visual concepts in images [3] or joint occurrences of local descriptors and tags [10]. Also, multiple instance learning methods have been used to detect local features associated with a concept [19]. Image annotation is also performed at a large scale (see the 'Automatic Linguistic Indexing of Pictures - Real Time' [ALIPR] server [6]). Closest to ours, however, is the work by Fergus et al. [2], who introduced the idea of learning from low-quality online data for the domain of images.

If dealing with video content, the detection of semantic concepts often follows a keyframe extraction for efficiency reasons, leading to an image retrieval problem (e.g., [10, 18]). A valuable source of information beyond such static image content is *motion*, which has for example been employed in form of motion descriptors [8].

(a) (b) (c) (d) (e)

Fig. 1: Some sample keyframes extracted from videos with the tag `desert`. Tagging such material is made difficult by the visual diversity of the concept (a,b,c), shots not directly visually linked to to the tag (d), and low production quality (e).

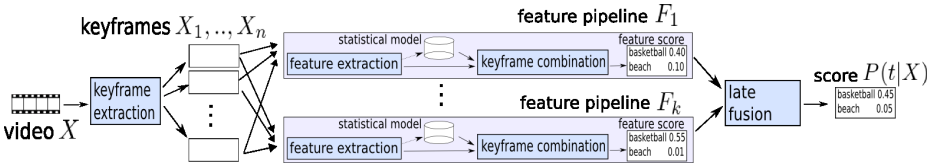


Fig. 2: An overview of our tagging system: a video X is represented by keyframes $X_1, .., X_n$. Each of them is fed to feature pipelines $F_1, .., F_k$, in which visual features are extracted and give scores $P_{F_j}(X_i)$ for each keyframe. These posteriors are fused over all keyframes, and finally over all features, to obtain the result $P(t|X)$.

As far as video annotation is concerned, a lot of related work has been done as part of TRECVID[3], an annual video retrieval contest that hosts quantitative evaluations on an extensive corpus of news video. In its "high-level features" task, the automatic tagging of shots is addressed. To boost standardization and comparability of results, groups share large sets of manual annotations [12], low-level features, and baseline results [15].

When dealing with online video content, several characteristics need to be respected: First, online video comes in a greater diversity, ranging from home video to commercial TV productions. Second, annotations in TRECVID are done on shot level, while we are interested in tagging whole videos. To the best of our knowledge, no prior research with the focus on online video tagging exists.

3 Our Approach

Given a video X and a semantic concept t (in the following referred to as a "tag"), the problem of automatic annotation is to return a "score" $P(t|X)$, i.e. the probability that the tag is present in the video.

Figure 2 gives an overview of our system architecture: we represent a video X by a set of representative keyframes $X_1, .., X_n$. Each keyframe X_i is fed to several "feature pipelines" $F_1, .., F_k$, each using visual features of a certain type to return a score $P_{F_j}(t|X_i)$. These pieces of evidence are first fused over all keyframes of a video, obtaining feature-specific scores $P_{F_j}(t|X)$. Second, those are again fused over all feature pipelines in a late-fusion step, obtaining the final score $P(t|X)$. In the following, the system components are described in the order of processing.

3.1 Keyframe Extraction

A standard way to extract a set of representative keyframes for each video is to segment the video into shots and use one frame per shot as a keyframe. This causes considerable information loss for long shots containing strong camera motion and scene activity, which is why an adaptive approach providing multiple keyframes per shot seems more adequate for online videos.

[3] http://www-nlpir.nist.gov/projects/t01v/

We use a divide-and-conquer approach that delivers multiple keyframes per shot in two steps: first, shot boundary detection is applied, for which reliable standard techniques exist [7]. Second, for each of the resulting shots, a clustering approach is applied similar to [11]: we extract MPEG-7 color layout descriptors [9] for all frames in a shot and then fit a Gaussian mixture model to the resulting feature set using k-means. For each mixture component, the frame next to the center is extracted as a keyframe. The number of components is determined using the Bayesian Information Criterion (BIC) [13].

3.2 Feature Pipelines

A key aspect of our tagging system is the combination of several visual features. We organize these in "feature pipelines" $F_1, .., F_k$, each of which represents a type of visual feature (e.g., color histograms) and gives a feature-specific score $P_{F_j}(t|X_i)$ for a keyframe X_i. We use three feature pipelines outlined in the following. Note, however, that more pipelines can be integrated easily.

Pipeline 1 - Color and Texture: This feature pipeline uses frame-level descriptors based on color (RGB color histograms with $8 \times 8 \times 8$ bins) and texture (Tamura features [16]). Both features are combined by early fusion (i.e. concatenated) to obtain a joint feature vector $F_1(X_i)$.

As a statistical model, we use nearest neighbor matching as illustrated in Figure 3: given a training set of tagged keyframes Y, we find the nearest neighbor $x' := \arg\min_{y \in Y} ||F_1(y) - F_1(X_i)||_2$, and the score for a tag t is a vote for the tag of this neighbor:

$$P_{F_1}(t|X_i) := \delta(t, t(x')) \tag{1}$$

Pipeline 2 - Motion: Some semantic concepts (e.g., `interview`) can be characterized better by a discriminative motion pattern than by color or texture. To do so, we use a simple compressed domain feature of block motion vectors extracted by the MPEG-4 codec XViD[4] to describe *what* motion occurs as well as *where* in the frames it occurs.

For this purpose, the spatial domain is divided into 4×3 regular tiles, and for each tile a two-dimensional 7×7 histogram over the 2D components of all motion vectors in a shot is stored (vectors are clipped to $[-20, 20] \times [-20, 20]$). By concatenating all those histograms, a 588-dimensional descriptor is extracted on shot level, i.e. it is the same for all keyframes in a shot.

Like for color and texture, nearest neighbor matching is used to model the keyframe score.

Pipeline 3 - Visual Words: Modern recognition systems have been successful by representing images with collections of local image regions (or "patches", respectively). These patch-based techniques achieve excellent robustness with

[4] www.xvid.org

Fig. 3: Left: In nearest neighbor matching, a test frame (top row) votes for the tag of its nearest neighbor in the training set (bottom row). **Right**: The maximum entropy model learns these sample patches as discriminative for the tag `eiffeltower`.

respect to partial occlusion, deformations, and clutter, which is why we adapt a similar approach for our system.

More precisely, we use a "bag-of-visual-words" representation [1, 2, 14]. This model clusters visual features according to their appearance into patch categories referred to as "visual words". Histograms over these visual words indicate the frequency with which all kinds of features appear in a frame, an analogy to the "bag-of-words" model from textual information retrieval.

A vocabulary of 500 visual words was learned by sampling patches of size 32× 32 pixels at regular steps of 16 pixels and clustering them using k-means. Patches are described by low-frequency discrete cosine transform (DCT) coefficients in YUV space. We extract 36 coefficients for the intensity, and 21 for each chroma component in a zigzag pattern, obtaining a 78-dimensional patch descriptor.

As a statistical model for the resulting 500-bin histograms, we adapt a discriminative approach based on the maximum-entropy principle, which has successfully been applied to object recognition before [1]. The posterior is modeled in a log-linear fashion:

$$P_{F_3}(t|X_i) \propto \exp\left(\alpha_t + \sum_{c=1}^{500} \lambda_{tc} h_i^c\right), \tag{2}$$

where h_i^c is entry number c in the visual word histogram for frame X_i. The parameters $\{\alpha_t, \lambda_{tc}\}$ are estimated from a training set of tagged frames using an iterative scaling algorithm [1].

3.3 Fusion

From several keyframes of the input video and from several feature pipelines, we obtain many weak pieces of evidence in form of scores indicating the presence of semantic concepts. These are fused in two steps to obtain a global score.

Keyframe Fusion: For a fusion over the keyframes $X_1, .., X_n$ of a video X, we use the well-known *sum rule* from classifier combination:

$$P_{F_j}(t|X) = \frac{1}{n} \sum_{i=1}^{n} P_{F_j}(t|X_i) \tag{3}$$

Late Fusion: To combine scores obtained from several feature pipelines, several standard measures from classifier combination can be applied (like the sum rule, product rule, etc.). We present a small study in the following experimental section (Figure 5), in which we evaluate several combination strategies.

4 Experiments

We present quantitative experiments on real-world online videos to study (1) how the single components of the system influence its overall performance, and (2) how the system can be adapted to different kinds of video data.

Most of our experiments are done on a database of real-world online videos we downloaded from the video portal `youtube.com`. We selected 22 tags manually, including activities (e.g., `riot`, `sailing`), objects (e.g., `cat`, `helicopter`), and locations (e.g., `desert`, `beach`). For the complete list of tags, please visit our website http://demo.iupr.org/videotagging/tagging-description.html.

We used the `youtube` API to download 100 videos per tag (total database: 2200 videos / 194 hrs.). The whole set was separated into a training set (50 videos per tag), a validation set, and a test set (both 25 videos per tag). To avoid training on the testing data, the test set was only used for the final evaluation.

A problem for the evaluation are duplicate or slightly edited videos uploaded multiple times by different users. We identify and remove (near-)duplicates in two steps: First, exact duplicates are detected automatically using a signature matching similar to [4]. Second, near-duplicates are removed by a manual check of videos that gave suspiciously good results in our tagging experiments.

4.1 Feature Modalities

Figure 4 illustrates the influence of several feature pipelines on the performance on the system (all fusion of features was done using the sum rule): when starting only with color and texture (feature pipeline 1), a mean average precision

feature comb.	MAP
Color and Texture (CT)	25.4
Motion (M)	17.9
CT+M	27.1
Visual Words (VW)	29.0
CT+M+VW	34.2

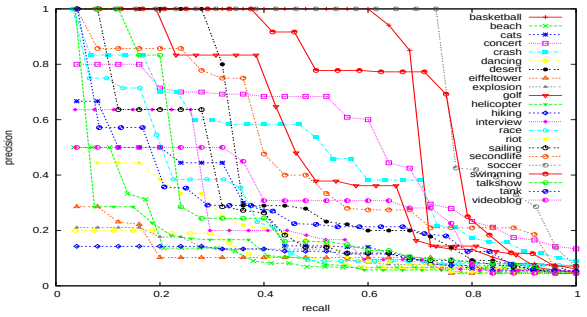
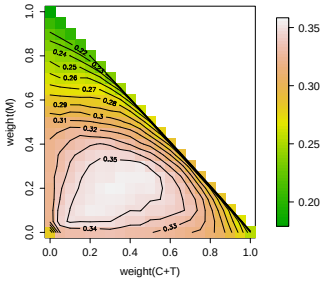


Fig. 4: Left: Experimental results for several feature combinations in terms of mean average precision (MAP). **Right**: The recall-precision curves when fusing all features. The average precision per concept varies between 81% (`soccer`) and 11% (`hiking`).

fus. strat.	MAP (%)
`sum`	34.2
`prod`	19.6
`min`	19.6
`max`	34.0
`wsum`	35.5



feature combination	MAP (%) `adapt`	`first`	`reg`
Color and Texture (CT)	25.4	16.8	29.4
Motion (M)	17.9	16.6	16.2
CT+M	27.1	21.7	24.7
Visual Words (VW)	29.0	25.4	28.2
CT+M+VW	34.2	31.6	34.8

Fig. 5: Left: Tagging performance when using several combination strategies. **Center**: The MAP on the test set plotted against the weights for color+texture and motion. The sum rule $(0.33, 0.33)$ and the weights learned from the validation set $(0.45, 0.2)$ are both near to the optimal peak $(0.3, 0.2)$. **Right**: Comparing our keyframe extraction (`adapt`) with two baselines.

(MAP) of 25.4% is achieved. Though motion (feature pipeline 2) alone does not improve performance, it supports the system when combined with color and texture (MAP 27.1%). An in-depth analysis reveals the strongest improvements for motion-affine concepts like `videoblog` (from 17 to 37%) and `riot` (from 14 to 23%).

Visual words by themselves (feature pipeline 3) give an even better performance, particularly for objects like `eiffeltower` (here, the MAP was improved from 7.3% to 70.6%). This is because the maximum-entropy model associates patches with a weight and thus emphasizes discriminative features of an object, as is illustrated in Figure 3. Finally, a sum rule fusion of all features achieves the best overall performance.

We also compared the sum rule fusion to several other classifier combination strategies. (Figure 5, left). The sum rule performs superior to other methods, which agrees with earlier results [5] that claim good robustness properties against noise in the weak (in our case, keyframe) estimates. Our results confirm that this robustness is crucial in our context, since many keyframes may not be visually related to the true tag and thus give misleading scores.

We also tested a more general approach, namely a *weighted sum*:

$$P(t|X) = \sum_{j=1}^{3} w_j P_{F_j}(t|X), \tag{4}$$

where the feature weights $(w_1, w_2, w_3) = (0.45, 0.2, 0.35)$ are learned from the validation set (i.e. color and texture are given the highest weight, and motion the lowest). The left Table in Figure 5 indicates that by learning feature weights from the validation set, we obtain a slightly better performance with 35.5%.

An in-depth view of the performance for all 22 tags is given in form of recall-precision curves in Figure 4. Obviously, a successful tagging strongly depends on the concept: Sports like `soccer` and `swimming` are easy to tag due to their restricted color layout and low diversity in appearance, while concepts with a high visual diversity are difficult, like `explosion` or `dancing`.

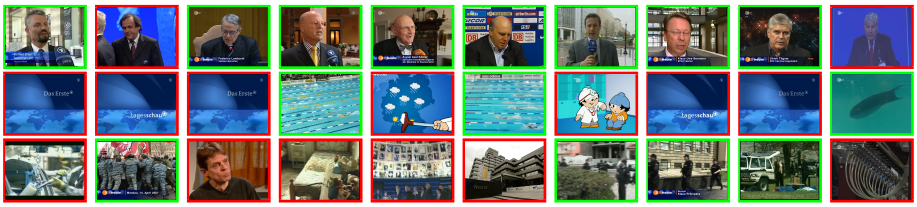


Fig. 6: Shots from a TV news dataset for which our system returns the highest scores, for the concepts `interview`(top), `swimming`(center), and `riot`(bottom). True positives are highlighted in green, false positives in red.

4.2 Keyframe Extraction

In this experiment, we compare our keyframe extraction (Section 3.1) with two baseline methods to investigate whether our adaptive approach is in fact essential for a successful tagging. The first baseline named `first` uses only the first frame of a shot as a keyframe (which is often done in practice). It generates only 56% keyframes compared to ours (ca. 97.000). The right table in Figure 5 shows that the use of additional keyframes leads to performance increases between 2 and 9% – i.e., tagging is improved by using multiple keyframes per shot.

Note that our clustering approach adapts to the activity of a shot, i.e. it produces more keyframes if the content of a shot varies. To answer whether this adaptivity is essential for tagging, we compare our approach to a second baseline (`regular`) that regularly samples keyframes at an interval of about 7 seconds. This baseline generates more keyframes than our approach (8.2%), and does not adapt to the content of a shot. The table shows about the same performance as for our adaptive method. This result indicates that the use of adaptive keyframes plays a negligible role.

4.3 Generalization to Different Video Data

Obviously, our system makes use of redundancy in youtube videos, which may occur due to duplicate shots or series sharing common production styles and locations. While we explicitly eliminate the former, our system still implicitly uses the latter, weaker type as is illustrated by the rightmost match in Figure 3 (left). While this helps the system to tag videos from the online portal trained on, it is unclear how the system generalizes to different content. This is why we tested our system trained on `youtube` on two other data sources.

TV News Data: This test set contains 5.5 hours of unlabeled German news video (ca. 4.000 keyframes). Given a tag, our system (all three feature pipelines, sum rule fusion) returns the TV shots sorted by their scores. In Figure 6, we illustrate the shots with top scores for three tags we expect to occur in news video. For `interview`, the system gives a near-perfect result (only 2 false positives). For `swimming`, many false positives can be observed, since the system is attracted by blue background that has not been present in the training set. Finally, we obtain four hits for the concept `riot`.

training set	test set	
	youtube	revver
youtube	37.0	11.1
revver	15.7	31.4
both	33.6	26.4

Fig. 7: Left: Sample frames for the concept `crash` from `youtube` (top) and `revver` (bottom). While `youtube` videos show mostly car and motorbike crashes, `revver` contains also skiing and biking accidents. **Right**: Our results show that a joint tagger for multiple video portals (here, `youtube` and `revver`) can be successful if trained on all data sources.

Though – due to the lack of ground truth – no quantitative results can be presented. , this result generally indicates that tagging of TV data can be learned from online video data.

Revver Video Portal: For this experiment, we use videos from the portal `revver.com`. We created a dataset similar to the `youtube` one. Two concepts were left out, and for four other concepts less than 100 videos were found. We used the system setup that gave the best results on the `youtube` data (all features, sum rule fusion). Since no further parameter tuning was done, we split the `revver` data into a $\frac{2}{3}$ training set and $\frac{1}{3}$ test set.

Figure 7 illustrates that the system generalizes poorly if trained on one portal and applied to the other. An explanation for this is illustrated in Figure 7 for the example of the concept `crash`: `youtube` videos (top row) contain lots of TV material showing car and motorbike races, while `revver` videos show significantly more home video content (here, skiing or biking). Obviously, a system that is not trained on this novel content cannot correctly tag such data.

Therefore, we studied if a generalized tagger can be created by training the system on *both* data sources. We obtain a system that performs comparable (about 4% worse) to the specialized systems trained for the single portals. This demonstrates how a general tagger can be created by adapting the training set.

5 Conclusions

In this paper, we have introduced a system that learns to detect semantic concepts in videos by watching content from online video portals. Our experimental results show that fair tagging results can be obtained when combining several visual feature modalities, namely color, texture, motion, and a patch-based approach.

However, two key aspects of learning from online videos have been neglected so far: (1) Can training be adapted better for certain concept types (e.g., objects vs. locations)? (2) Can the system be made robust to shots in a video that are irrelevant for a concept? So far, our answer to both questions has been the integration of multiple feature modalities using a robust sum rule fusion. We expect a better tagging by addressing these problems using explicit models, and therefore envision our system to be a baseline for future work.

6 Acknowledgements

This work was supported in part by the Stiftung Rheinland-Pfalz für Innovation, project InViRe (961-386261/791).

References

1. Deselaers T., Keysers D., Ney H., "Discriminative Training for Object Recognition Using Image Patches", *CVPR*, pp. 157-162, Washington, DC, 2005.
2. Fergus R., Fei-Fei L., Perona P., Zisserman, A., "Learning Object Categories from Google's Image Search", *Computer Vision*, Vol. 2, pp. 1816-1823, 2005.
3. Barnard K., Duygulu P., Forsyth D., de Freitas N., Bleib D., Jordan M., "Matching Words and Pictures", *J. Mach. Learn. Res.*, Vol. 3, pp. 1107-1135, 2003.
4. Hoad T.C., Zobel J., "Detection of Video Sequences using Compact Signatures", *ACM Trans. Inf. Systems*, Vol. 24, No. 1, pp. 1–50, 2006.
5. Kittler J., Hatef M., Duin R., Matas J., "On Combining Classifiers", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 20, No. 3, pp. 226-239, 1998.
6. Li J., Wang J., "Real-time Computerized Annotation of Pictures", *Intern. Conf. on Multimedia*, pp. 911-920, Santa Barbara, CA, 2006.
7. Lienhart R., "Reliable Transition Detection in Videos: A Survey and Practitioner's Guide", *Int. J. of Image and Graphics*, Vol. 1, No. 3, pp. 469-286, 2001.
8. Ma Y.-F., Zhang H.-J., "Motion Pattern-based Video Classification and Retrieval", *EURASIP J. Appl. Signal Proc.*, No. 1, pp. 199–208, 2003.
9. Manjunath B.S., Ohm J.-R., Vasudevan V.V., Yamada A., "Color and Texture Descriptors", *IEEE Trans. on Circuits Syst. for Video Techn.*, Vol. 11, No. 6, pp. 703–715, 2001.
10. Feng S.L., Manmatha R., Lavrenko V., "Multiple Bernoulli Relevance Models for Image and Video Annotation", *CVPR*, pp. 1002-1009, Washington, DC, 2004.
11. Hammoud R., Mohr R., "A Probabilistic Framework of Selecting Effective Key Frames for Video Browsing and Indexing", *Intern. Worksh. on Real-Time Img. Seq. Anal.*, pp. 79-88, Oulu, Finland, 2000.
12. Naphade M., Smith J., Tesic J., Chang S.-F., Hsu W., Kennedy L., Hauptmann A., Curtis J., "Large-Scale Concept Ontology for Multimedia", *IEEE Multimedia*, No. 13, Vol. 3, pp. 86-91, 2006.
13. Schwarz G., "Estimating the Dimension of a Model", *Ann. of Stat.*, No. 6, Vol. 2, pp. 461–464, 2003.
14. Sivic J., Zisserman A., "Video Google: A Text Retrieval Approach to Object Matching in Videos", *ICCV*, pp. 1470-1477, Washington, DC, 2003.
15. Snoek C. et al., "The MediaMill TRECVID 2006 Semantic Video Search Engine", *TRECVID Workshop* (unreviewed workshop paper), Gaithersburg, MD, 2006.
16. Tamura H., Mori S., Yamawaki T., "Textural Features Corresponding to Visual Perception", *IEEE Trans. on Sys., Man, Cybern.*, No. 6, Vol. 8, pp. 460-472, 1978.
17. Ulges A., Schulze C., Keysers D., Breuel T., "Content-based Video Tagging for Online Video Portals", *MUSCLE/ImageCLEF Workshop*, Budapest, 2007.
18. Snoek C., Worring M., van Gemert J., Geusebroek J.-M., Smeulders A., "The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia", *Intern. Conf. on Multimedia*, pp. 421-430, Santa Barbara, CA, 2006.
19. Yang C., Lozano-Perez T., "Image Database Retrieval with Multiple-Instance Learning Techniques", *Int. Conf. on Data Eng.*, pp. 233-243, San Diego, CA, 2000.