In-Video Product Annotation with Web Information Mining

Guangda Li
National University of Singapore,
Meng Wang
Hefei University of Technology,
Zheng Lu
National University of Singapore,
Richang Hong
Hefei University of Technology,
Tat-Seng Chua
National University of Singapore

Product annotation in videos is of great importance for video browsing, search and advertisement. However, most of the existing automatic video annotation research focuses on the annotation of high-level concepts, such as events, scenes and object categories. This paper presents a novel solution to the annotation of specific products in videos by mining information from the Web. It collects a set of high-quality training data for each product by simultaneously leveraging Amazon and Google image search engine. A visual signature for each product is then built based on the bag-of-visual-words representation of the training images. A correlative sparsification approach is employed to remove noisy bins in the visual signatures. These signatures are used to annotate video frames. We conduct experiments on more than 1,000 videos and the results demonstrate the feasibility and effectiveness of our approach.

Categories and Subject Descriptors: H.3.1 [Content Analysis and Indexing] indexing methods; I.2.10 [Artificial Intelligence] Video and Scene Understanding

General Terms: Algorithm, Design, Experimentation

Additional Key Words and Phrases: product annotation, video search, web mining

1. INTRODUCTION

With the rapid advances in storage devices, networks and compression techniques, video data from different domains are growing at an explosive rate. Video annotation (also widely known as video
concept detection or high-level feature extraction), which aims to automatically assign descriptive concepts to video content, has received intensive research interests over the past few years [Natsev et al. 2007][Smeaton et al. 2006]. However, most existing works on automatic video annotation focus on high-level concepts, such as events (e.g., airplane crash and running), scenes (e.g., sundown and beach) and object categories (e.g., car and screen) [Kennedy 2006], while a few research efforts focus on annotating specific product concepts, such as Iphone and Kindle. Product is a type of “object”, and the difference is that it is more specific. For example, in WordNet [Miller 1995] ontology, “product” is in the tree of “object”. There are also coarse-grained and fine-grained product concepts. For example, "Canon SLRs" can be a coarse-grained product concept while Canon 40D can be a finer-grained product concept. In this work, we mainly focus on the annotation of fine product concepts, namely, specific product concepts. The annotation of specific product concepts is of great importance to many applications such as video browsing, search and advertising. First, existing investigation on the query log of web video search shows that users use specific queries more frequently than general concepts [Wang et al. 2008] (such as those used in TRECVID). Second, specific product annotation is able to significantly improve the relevance of video advertising. Mei and Hua [2010] and Guo et al. [2009] have investigated context-based video advertisement with content analysis. If we can accurately predict whether a video clip or a frame contains a specific product, clearly the advertising relevance will be significantly boosted.

However, the automated annotation of products in videos is not an easy task. In comparison with general concepts, the automated annotation of products in videos has the following challenges. The first challenge lies on the training data. Learning-based video annotation approach heavily relies on the quality of training set, but manually collecting training samples is time-consuming and labor intensive. Besides, products are too numerous and new products keep emerging, hence in most case, it is hard to pre-train the model, and most of training needs to be done on the fly. Thus, we need to develop techniques to quickly collect lots of good quality training samples.

The second challenge is that there is a multi-view problem for products; that is, a specific product usually has different views, such as frontal, side and back views, and these views can be quite different visually. Therefore, we need to collect training samples those are descriptive for different views of a product.

The third challenge is the visual representation problem. Bag of Visual Words (BoVW) feature [Sivic and Zisserman 2003] is the most popular approach and has demonstrated its effectiveness in many applications, such as image classification, clustering, and retrieval. It first extracts Scale Invariant Feature Transform (SIFT) descriptors [Lowe 2004] on several detected keypoints or by densely sampling patches of each image. It then quantizes the descriptors into visual words. A BoVW histogram is generated to describe each image. However, whether we adopt keypoint detection or regularly sampling patches, the descriptor describes the whole image but not the product parts. This means that the BoVW representation is actually fairly noisy for product annotation.

In this work, we propose a scheme that is able to address the above two problems. Our scheme introduces two stages. In the first stages, our method automatically collects positive training images from the Internet by mining Amazon1, which is arguably the largest online shopping website and Google image search engine2. Given a product, we use the product name as query to collect a set of images from Amazon that usually describes the product with different views, as shown in Figure 1(a). But these images are too few for constructing a good model. Therefore, we utilize Google image search engine to “expand” the example images from Amazon. We again use the product name as text query to Google image search to collect a large set of images as shown in Figure 1(b). For each Amazon image,
we then collect its nearest neighbors through visual matching in the Google image search results. In this way, we can obtain a set of good quality training images for the product.

The second stage involves the training of a classifier on the fly. The conventional video annotation methods usually train classifiers based on several positive and negative samples. However, for product concepts, negative samples are usually much more than positive samples and they distribute in much broader domains. This introduces difficulty in training discriminative classifiers. In addition, we employ a very large visual codebook in order to enhance the discriminative ability of visual representations since for specific object retrieval, a large visual codebook will increase the precision [Philbin et al. 2007]. Besides, recent studies demonstrate that, when dealing with very high-dimensional feature space, directly averaging positive samples to generate a template for annotation is an effective choice [Zhou et al. 2009]. Therefore, we merge the BoVW histograms of these images to form a visual signature of the product, which embeds the visual information of different views and poses of the product. However, as previously introduced, the representation constructed in this way is actually fairly noisy. In this paper, we propose a correlative sparsification approach to reduce noises. It is formulated as a regularization framework that contains three terms. The first term keeps the obtained signatures to be close to the original ones. The second term minimizes the L1-norm of the obtained visual signatures, which is able to enforce them to be sparse. The third term is graph Laplacian that keeps the signatures of similar products close. The final sparsified signatures can be utilized to predict whether a video frame contains a product or not. The overall process is quite similar to the signature file in document retrieval, which is used to create a quick and dirty filter to discriminate relevant and irrelevant documents [Zobel et al. 1998]. Thus, we regard this accumulated BoVW feature of a product as its visual signature.

The main contributions of this paper can be summarized as follows.

(1) We propose an in-video product annotation scheme. To the best of our knowledge, this is the first work dedicated to investigating product annotation in videos.

(2) To cater the huge variety of frequent introduction of new products, we introduce a simple yet effective method to collect product visual examples from the web. The whole process is automated and does not require humans’ manual efforts.

(3) We introduce a novel correlative sparsification method to generate the sparse visual signatures of products. It is able to reduce the noise of the visual signatures, such that better annotation
performance can be achieved. Using L1-norm constraint to enforce sparseness is a widely-applied strategy, but the correlative sparsification is a novel method.

The rest of the paper is organized as follows. In Section 2, we provide a short review of the related work. In Section 3, we give an overview of our approach. In Section 4, we introduce the detailed product image collection approach, including Amazon example collection and their expansion via Google image search engine. In Section 5, we introduce the detailed in-video product annotation approach, including the visual signature generation and video frame relevance estimation. Experimental results are presented in Section 6. Finally, we conclude the paper in Section 7.

2. RELATED WORK

2.1 Video Annotation

The National Institute of Standards and Technology (NIST) has established the “high-level feature extraction” task in TREC video retrieval evaluation (TRECVID) from 2002\(^3\). It aims to assign each video clip a set of relevant concepts. Extensive research efforts have been dedicated to this task [Natsev et al. 2007] [Smeaton et al. 2006] [Snoek and Worring 2009] [Wang et al. 2009]. However, the concepts investigated in TRECVID, such as those in LSCOM [Naphade et al. 2006], mainly focus on object categories, scenes and events, whereas highly specific objects, such as products, are overlooked. This can be attributed to the fact that annotating products is still a very challenging task because of the difficulty in selecting enough training data and generating appropriate visual representations for a single product. Actually for many simple object categories, such as chair and telephone, the annotation performance is still not satisfactory (the best AP results for these two concepts are below 0.3) [Tang et al. 2008] [Wang et al. 2009].

There are already some research efforts that attempt to automatically mine training data for image or video annotation from the web. Schroff et al. [2011] proposed a multi-model approach to re-rank the images returned by Google image search engine. Li et al. [2007] proposed an incremental learning approach to collect online images. Ulges et al. [2010] described a scheme that employs online video portal as a data source for learning concept detector. Setz and Snoek [2009] investigated the possibility of using social tagged images as a training resource for concept-based video search. However, for products, the quality of the training data obtained by these methods may not be good enough as they may contain too much noise. The obtained images also may not cover many different views of the products. Our proposed method investigates Amazon, which can be viewed as a corpus containing most popular products that is built with expert knowledge. The Amazon examples are then used to filter and select more relevant images from the Google image search results to expand the training data with better quality images.

2.2 Product Search

The literature regarding product related image or video search is still relatively sparse. Jing and Baluja [2008] applied a PageRank-like algorithm based on visual links between images to improve the ranking performance for product image search. Xie et al. [2008] proposed client-server architecture for mobile device to undertake multi-modality search, one of which is content-based product retrieval by queries from phone camera. For commercial applications, Google goggles\(^4\) and Amazon SnapTell\(^5\) are able to provide users product search results by simply capturing a view using mobile phone. VideoSurf\(^6\)

\(^{3}\)TREC video retrieval evaluation: http://www-nlpir.nist.gov/projects/trecvid/
\(^{4}\)Google Goggles: http://www.google.com/mobile/goggles/
\(^{5}\)SnapTell: http://www.snaptell.com/
\(^{6}\)VideoSurf: http://www.videosurf.com/
can provide better video search performance based on video content analysis. However, there is very little research on product annotation in videos, and our work well complements the existing efforts.

3. OVERALL FRAMEWORK

The main scheme of our approach is illustrated in Figure 2. There are two stages to detect products within videos, namely the product visual signature generation stage and video processing stage. The visual signature generation stage mainly consists of three components: collection of visual example from Amazon, expansion of examples with Google image search results, and generation of visual signature from training examples. The detailed process is as follows. Given a product name, we first use the name to collect the associated images on Amazon. We then collect Google image search results by using the product name as query. We use each Amazon example to re-rank the Google image search results using visual matching and the $n$-nearest neighbors are collected. In this way, we obtain $kn$ positive examples for the product, where $k$ is the number of Amazon examples. The on-line video processing stage consists of two components, feature extraction and automated product annotation in videos. From a given video stream, we identify a set of keyframes, and for each keyframe, we extract the SIFT features and generate the BoVW histogram accordingly. The product annotation in videos is thus accomplished by comparing the visual signature of each product with the BoVW histogram.

4. MINING PRODUCT IMAGES FROM INTERNET

In this section, we detail our on the fly training data collection component. For a product, generally we can collect a set of high-quality examples from Amazon and these example images are usually able to cover different views of the product. However, these images are too few for constructing a good visual signature for the product (the numbers of examples collected from Amazon usually vary from 1 to 8). On the other hand, there are plenty of images in different sizes and views available on the Internet, which can be easily accessed through image search engines. But the images contain a lot of noise as they are indexed by text information (such as title and surrounding text) and thus many of them are irrelevant to the query. Here, we utilize the product images from Amazon as seeds to filter out noisy web images of the product, and it can also be regarded as the process of expanding the product images from Amazon using the web image database. For each Amazon image, we collect its neighbors in the Google image search results. In this way, we can obtain a set of positive training images for the product.
Guangda Li

Fig. 3. The schematic illustration of training data collection process. For each example image on Amazon, its nearest neighbors in Google image search results are collected. It can be regarded as the process of expanding Amazon examples with Google image search engine.

As shown in Figure 3, the product image expansion process works as follows. First, given a specific product name, such as Canon G9, we use this as query to collect product images from Amazon and crawl web images from Google image search engine. Second, we extract the BoVW feature of all images. After that, we collect the $n$ nearest neighbors for each Amazon example. Here we employ the intersection similarity measure, which is defined as:

$$
sim(x, y) = \frac{\sum_{d=1}^{D} \min\{(x_d), (y_d)\}}{\min\{\sum_{d=1}^{D} x_d, \sum_{d=1}^{D} y_d\}}
$$

where $x$ and $y$ are two BoVW histograms and $D$ is the length of the histogram. In this way, we obtain $kn$ positive training images in total, where $k$ is the number of Amazon examples.

5. IN-VIDEO PRODUCT ANNOTATION

In this section, we present the proposed in-video product detection in details. First we introduce the scheme of visual signature generation for each product by utilizing our correlative sparsification method. Next, we describe the estimation of similarity measure between different products, which is used in the correlative sparsification approach. Finally, we introduce the relevance score estimation of video frames for a certain product.

5.1 Visual Signature Generation

Conventional video annotation methods usually regard the annotation of each concept as a binary classification task, and discriminative classifiers, such as SVM, are frequently employed. But for product annotation in videos, negative samples are usually much more than positive samples and they are also distributed in much broader feature space. This makes it difficult to train discriminative classifiers with both positive and negative samples. However, recent studies demonstrate that, when dealing with very high-dimensional feature space, directly averaging positive samples to generate a template
for annotation is an effective choice [Zhou et al. 2009]. Hence we first merge the visual representation from multiple positive example images to generate an accumulated histogram for each product. Since there are many noises caused by the descriptors from image background, there are actually many noisy bins in the accumulated histogram. One approach to reduce the noise is to adopt sparsification, which fits the L1-regularized least square optimization problem [Kim et al. 2007]. Here we call it L1-sparsification:

$$\arg \min_{v_i} \|v_i - \bar{v}_i\|^2_2 + \lambda_1 \|v_i\|_1$$  \hspace{1cm} (2)

where $\|\cdot\|_2$ and $\|\cdot\|_1$ indicate the L2-norm and L1-norm respectively. The first term keeps the obtained signatures to be close to the original ones. The second term minimizes the L1-norm of the obtained visual signatures, which makes the signatures sparse. The parameter $\lambda_1$ modulates the effect of L1-norm, $\bar{v}_i$ is the original accumulated BoVW histogram for the $i$-th product, and $v_i$ is the to-be-learned visual signature.

Meanwhile, we observe that several products of same class have close appearances. For example, the products Canon 40D and Nikon D90 have very close appearances. Thus, the histogram representation of these two products should be very close. Therefore, we add a graph Laplacian term to the above formulation that can keep the visual signatures of similar products close, and thus the formulation becomes:

$$\arg \min_{\{v_1, v_2, \ldots, v_n\}} \sum_{i=1}^{n} \|v_i - \bar{v}_i\|^2_2 + \lambda_1 \sum_{i=1}^{n} \|v_i\|_1 + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \|v_i - v_j\|^2_2$$  \hspace{1cm} (3)

where $w_{ij}$ is the similarity between products $i$ and $j$, and $\lambda_2$ is the parameter that modulates the effect of the graph Laplacian term. We can see that, the graph Laplacian term, $\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \|v_i - v_j\|^2_2$, actually joins the signatures of all products. This means that we will generate all the signatures correlatively. Such principle has been widely investigated in multi-task learning and many applications such as annotation, retrieval and codebook generation [Li et al. 2010][Gao et al. 2010][Geng et al. 2008].

Directly solving Eq.3 is difficult and here we adopt an alternate optimization approach. We consider all the signatures except $v_i$ as fixed, and thus the formulation becomes:

$$\arg \min_{v_i} \|v_i - \bar{v}_i\|^2_2 + \lambda_1 \|v_i\|_1 + \lambda_2 \sum_{j=1}^{n} w_{ij} \|v_i - v_j\|^2_2$$  \hspace{1cm} (4)

We re-write the optimization problem as

$$v_i = \arg \min_{v_i} \| \begin{pmatrix} I \sqrt{\lambda_2 w_{11}} \vdots \sqrt{\lambda_2 w_{1n}} \\ \vdots \vdots \vdots \\ \sqrt{\lambda_2 w_{n1}} I \sqrt{\lambda_2 w_{12}} \vdots \sqrt{\lambda_2 w_{n2}} \vdots \sqrt{\lambda_2 w_{nn}} I \end{pmatrix} v_i - \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \|_2^2 + \lambda_1 \|v_i\|_1$$  \hspace{1cm} (5)

where $I$ is the $D \times D$ identity matrix. It thus turns into a L1-regularized least square optimization problem. We solve it using interior-point method [Kim et al. 2007]. From the above equation, we can easily derive an iterative process to solve each $v_i$ by repeatedly updating them. Figure 4 illustrates the process. Since the objective function in Eq.5 is lower bounded by 0 and decreases for each step, the convergence of the process is guaranteed.
5.2 The Estimation of Inter-Product Similarity

We estimate the similarity of two products based on their visual examples. Figure 5 shows three example sets of Canon 40D, Nikon D90, and Xbox respectively. Intuitively, we can see that the first two rows of images are visually much closer than the third row. Following [Wang et al. 2010], we define the visual similarity between two set of product images $\mathcal{P}_i$ and $\mathcal{P}_j$ as:

$$w_{ij} = \frac{1}{2|\mathcal{P}_i|} \sum_{k=1}^{|\mathcal{P}_i|} \max_{p \in \mathcal{P}_j} \text{sim}(p_i^{(k)}, p) + \frac{1}{2|\mathcal{P}_j|} \sum_{k=1}^{|\mathcal{P}_j|} \max_{p \in \mathcal{P}_i} \text{sim}(p_j^{(k)}, p)$$

where $|\mathcal{P}_i|$ and $|\mathcal{P}_j|$ are the numbers of the images in $\mathcal{P}_i$ and $\mathcal{P}_j$ respectively, $p_i^{(k)}$ indicates the $k$-th product image in the set $\mathcal{P}_i$, $p_j^{(k)}$ indicates the $k$-th product image in the set $\mathcal{P}_j$, and $\text{sim}(., .)$ is the similarity of a image pair from different sets. In the term $\max_{p \in \mathcal{P}_j} \text{sim}(p_i^{(k)}, p)$, we adopt the maximum over all possible instantiations of $p$ in $\mathcal{P}_j$. We can see that this similarity measure has the following properties:

ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 1, No. 1, Article 1, Publication date: January 2012.
In-Video Product Annotation by Web Information Mining

Fig. 6. The sketch of category hierarchy structure of Consumer Electronic domain in Amazon.

(1) \( w_{ij} = w_{ji} \), i.e., the similarity is symmetry.
(2) \( w_{ij} = 1 \) if \( P_i = P_j \), i.e., the similarity of two products is 1 if their image sets are identical.
(3) \( w(P_i, P_j) = 0 \) if and only if \( \text{sim}(p', p'') = 0 \) for every \( p' \in P_i \), and \( p'' \in P_j \), i.e., the similarity is 0 if and only if every pair formed by the two image sets has zero similarity.

For the above equation, we use the histogram intersection to calculate the similarity of two images. In addition, we also investigate the category knowledge on Amazon. We adopt a simple rule: if two products belong to different sub-categories\(^7\), we set their similarity measurement to 0.

5.3 Product Detection in Video by Relevance Estimation

The product relevance estimation for video frames is accomplished by comparing the visual signature of each product to the BoVW features of the frames. Here we still employ the histogram intersection. For a frame, its relevance score for \( i \)-th product is:

\[
s(f, v_i) = \frac{\sum_{d=1}^{D} \min\{f_d, v_{i,d}\}}{\min\{\sum_{d=1}^{D} f_d, \sum_{d=1}^{D} v_{i,d}\}}
\]

where \( f \) is the frame’s visual BoVW histogram, and \( v_i \) is the visual signature file for the \( i \)-th product.

6. EXPERIMENTS

In this section, we report our experiments. We first introduce our dataset and the products for annotation. We then compare different training data collection approaches and visual signature generation methods. We also investigate the integration of our approaches of automated product annotation in

\(^7\)Amazon has organized the products online with tree structure. There are some categories contains several sub-categories, such as Phone, Digital SLR and MP3 (details can be found on http://www.amazon.com/gp/site-directory). Figure 6 illustrates the tree structure of the Consumer Electronic category. Here we only use the similarity measurements of products in the same smallest sub-categories, and otherwise the similarity is set to 0.
videos with text metadata. The effects of parameters and the inter-product discriminative abilities of the visual signatures are analyzed as well.

6.1 Experimental Setting

In this work, we selected 20 popular products from the electronics domain for evaluation. They are Canon 40D, Nikon D90, Canon G9, Cisco 7960 phone, Blackberry 9700, Xbox360, Xbox Kinect, playstation3, Nintendo Wii, Amazon Kindle, Sony Vaio, Lenovo ThinkPad, Apple macbook pro, Casio hiking watch, Rolex Oyster watch, Sony NWZS754, Apple iPod touch 32 GB (4th Generation), Apple iPod nano 8GB (6th Generation), Panasonic Lumix DMCLX5, and Nikon Coolpix P7000. Figure 7 illustrates several example images for each product. We collected 1044 web videos from YouTube by issuing the above product names as queries in November 2010. Some videos have high quality and the contained products are highly typical (such as ads videos) and there are also several videos in which the products are not so typical. For each video, we extracted a keyframe every 5 seconds. We obtained 52,941 keyframes in total. Following the strategy in TRECVID, we annotated a frame as relevant to a product if it can be recognized no matter whether it is small and non-typical. Among the keyframes, there are 16,329 that are relevant to at least one of the products and 36,162 keyframes that are irrelevant to any product.

For feature representation, we employed Difference-of-Gaussian method to detect keypoints and from each keypoint we extracted 128-dimensional SIFT features [Lowe 2004]. The SIFT features were grouped into 160,000 clusters with hierarchical K-means [Nister and Stewenius 2006]. Therefore, each image is represented by a 160,000-dimensional BoVW histogram.

6.2 Experimental Results

6.2.1 On Content-based Product Image Expansion. To test the effectiveness of the content-based product image expansion by queries from Amazon, we compared it with the method that directly
collects training data from Google Image search engine in terms of the number of collected positive images. We keep the number of collected images to be 300. To save the labor cost, we randomly sample 100 images from each set for manual labeling. Figure 8 illustrates the numbers of positive examples collected by different methods for each product.

Clearly, our proposed approach that simultaneously integrates Amazon and Google image search results collects more positive training data than directly using the top results of Google image search results. This is because the Amazon images that we used to filter and select positive images from Google results cover diverse views and angles; and in many cases, we found that we are able to find positive images of certain views that are ranked very low in Google image search results up to the front. In addition, the positive images collected by our approach are able to retain the diversity of images returned by Google image search engine. This is because our Amazon images used for collecting positive images cover diverse views and angles. Figure 9 demonstrates the training data comparison for two products, Xbox Kinect and Nikon D90 respectively. In Figure 9(a) and Figure 9(c), we can see that Google can provide very diverse images but there are some unrelated images of the product. In Figure 9(b) and Figure 9(d), there are less unrelated images than the Google image search engine, and at the same time, our approach is able to retain the diverse characteristic of the images returned from Google. Later we will show that much better performance can be obtained by using the training data collected by our approach in comparison with only using Google image search engine.

6.2.2 On Visual Signature Generation. For performance evaluation metric, we adopted the well-known average precision (AP) to measure retrieval effectiveness [Smeaton et al. 2006]:

$$AP = \frac{1}{R} \sum_{j=1}^{S} \frac{R_j}{j} I_j$$  \hspace{1cm} (8)

where $R$ is the number of true relevant frames in a set of size $S$, $R_j$ is the number of relevant frames in the top $j$ results at any given index $j$, and $I_j = 1$ if the $j$th frame is relevant and 0 otherwise. Mean average precision (MAP) is the average of average precisions over all products. To comprehensively evaluate our approach, we first consider three choices of training data:
Using only Amazon examples. The images are thus very few.

(2) Using only top Google image search results. We use 300 top images for each product.

(3) The proposed approach that simultaneously integrates the Amazon and Google image search result. We use also the 300 top images for each product.

We denote the above three choices as “Amazon-only”, “Google-only” and “Amazon+Google”, respectively. For annotation algorithm, we consider the following three choices:

(1) directly using the accumulated BoVW histogram of all positive training images;

(2) using the L1-norm sparsification method, as shown in eg. 2; and

(3) using the correlative sparsification method as shown in eg. 3.

We denote the three choices as “Non-sparse”, “L1-norm sparsification”, and “Correlative sparsification”, respectively. For the second and third methods, the parameters $\lambda_1$ and $\lambda_2$ are empirically set to 5 and 0.05 respectively, which are shown to perform well in our experiments (the impacts of the two parameters will be investigated later).

We test the different combination of training data sources and annotation algorithms, and the MAP results are presented in Table I. From the results we can see that:

— The performance of “Google-only” is better than “Amazon-only”. This demonstrates that the example images on Amazon are too few to construct good visual signatures. It is true that Google images are
noisier than the Amazon images. But these Amazon images are too few for constructing a good model when we do not take any actions (Non-sparse). Though Google images are noisy, but there are still a lot of positive images, more than the number of Amazon images. This explains why Google-only performs better than the Amazon-only method.

— The “Amazon+Google” outperforms “Google-only”, and this confirms that effectiveness of our proposed approach.

— The performance of “L1-norm sparsification” is better than “Non-sparse”. This is because the sparsification approach reduces the noises of the BoVW histograms. The proposed “Correlative sparsification” further improves “L1-norm sparsification”, and this demonstrates the effectiveness of the graph Laplacian term. On the other hand, the Amazon+Google method only expands the Amazon images with Google. As long as the sparsification method is not used, the noisy bins in the model are still there, which will affect the accuracy of the relevance estimation. So it is reasonable that Google-only performs a little bit better than that of the Amazon+Google method as illustrated in the first line of the table. But this phenomenon doesn’t apply to the sparsification method because the sparsification method will keep the useful bins and eliminate the noisy bins very effectively.

— Figure 10 demonstrates an intuitive explanation. The sparsification methods are able to remove several noisy bins and thus the obtained visual signatures are better. The “Correlative sparsification” approach explores the correlation of multiple products and generates visual signatures with better quality.

6.2.3 On Multi-modal Product Annotation in Videos. We investigated the automated product annotation in videos by simultaneously integrating text and visual information, and see how much our proposed approach can help. We have collected the text information associated with the web videos, including their titles, descriptions and tags. We indexed the videos with the text and thus we can compute the relevance score of each video with respect to a product with BM25 model. We then compare the following two methods:

(1) Text only. We directly assign the relevance score of the whole video to its keyframes. For example, if Xbox 360 is contained in the related texts of a video, all key frames are regarded as relevant.

(2) Text + Visual. For those videos that contain the product name, we estimate the relevance scores of keyframes using our approach; for other videos, we set the relevance scores of their keyframes to 0.

Figure 11 illustrates the AP measurements obtained by the two methods for each product. We also further illustrate the AP measurements of using purely visual information for comparison (training data source is “Amazon+Google” and annotation algorithm is “Correlative sparsification”). From the results we can see that, purely text-based method only achieves a MAP of 0.23 and it is worse than the MAP of 0.39 achieved by using only visual information. By integrating text and visual information, the MAP measure can be boosted to 0.55. The results demonstrate the effectiveness of our visual signatures, and we can also see that integrating text and visual features can be a promising approach for automated product annotation in videos.
Fig. 10. BoVW sparsification can reduce the noisy visual words. The left column is the visual signature file of Amazon_rerank_Google by using No_sparse, L1_sparse and Corr_sparse respectively. The right column shows the corresponding visual words in a video frame. Due to privacy issue, we blurred the human face in the figure.

Fig. 11. The AP comparison of automated product annotation in videos with only using text clue, only using visual clue, and both text and visual clues. We can see that the results based on visual information are much better than those obtained using only text information. Combining text and visual information achieves the best results.
Fig. 12. Example annotated frames of product in video from YouTube video. We run the correlation sparsification method to generate Visual Signature Files for different products. The first row is the detection result for product Nikon D90, blackberry 9700, and Xbox Kinect, and the second row is the detection result for product Nikon Coolpix P7000, Apple Macbook, and Apple iPod touch 4th Generation.
6.2.4 On the Influence of Parameters. We also studied the sensitivity of the two parameters $\lambda_1$ and $\lambda_2$. Figure 13 illustrates the performance of product annotation in videos when we vary $\lambda_1$ and $\lambda_2$ respectively and with the other fixed. From the figures we can see that the annotation performance can be stable when $\lambda_1$ and $\lambda_2$ vary in a wide range (for example, $\lambda_1$ can vary from 1 to 10 and $\lambda_2$ can vary from 0.0005 to 0.05).

According to Eq. (3), we can see that the correlative sparsification method will degrade to the 1-norm sparsification method when $\lambda_2 = 0$. From Fig. 14 (b) we can see that the MAP measurement is 0.35 when $\lambda_2 = 0$, and this result is consistent with Table 2. We can observe from Fig. 14 (b) that the optimal value of $\lambda_2$ is fairly small. This is actually due to the fact that the scale of the third regularizer term in Eq. (3) is greater than the other two terms and thus $\lambda_2$ tends to be small. We can also perform a normalization for the three terms in the regularization formulation and then the problem can be solved.

6.2.5 On the Inter-Product Discriminative Ability. We also conducted a 20-way classification of the frames that are relevant to one of the 20 products to investigate the inter-product discriminative abilities of the visual signatures. We adopted a simple rule. For each frame, it is categorized to the product class that assigns its highest relevance score. The classification accuracy is 55.2% (note that the accuracy of a random 20-way classification will be only 5%). Figure 14 illustrates the detailed confusion matrix. We can see that, for several products, such as Amazon Kindle, Apple Ipod Nano and blackberry 9700, they can be easily distinguished. But the misclassification rates for several products such as Sony NWZS754 and Xbox Kinect are high.

6.3 Discussion

From the results shown in Figure 11, we can see that for several products we are able to achieve fairly good performance, but for some products, such as Xbox, Xbox Kinect and Playstation3, the AP measurements are below 0.2. The low performance is mainly due to the following reasons. First, for several products, the number of visual examples from Amazon is extremely few (only 1 or 2) and the Google image search results also contain many noises. The low-quality training data thus leads to non-satisfactory annotation performance. Second, visually similar products can also cause false detec-
Fig. 14. Confusion matrix for the 20-way classification problem. Classification rates for individual products are listed along the diagonal. The other entries are the misclassification rates, regarding different classes.

Table II. the number of the non-zero bins $D$ for each product at the stage of: (a) before sparsification, (b) after L1-sparsification, and (c) after correlative sparsification

<table>
<thead>
<tr>
<th>Product</th>
<th>$D$(Non-Sparse)</th>
<th>$M$(L1-Sparse)</th>
<th>$M$(Corr-Sparse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Kindle</td>
<td>6</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Apple iPod nano</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Apple iPod touch</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>CASIO hi-fi watch</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Lenovo ThinkPad</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Nikon Coolpix P7000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Panasonic Lumix DMC-LX5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>apple macbook pro</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>blackberry 9700</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>canan 40d</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>canan G9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>cisco 7960 phone</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>nikon D90</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>nintendo wii</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>playstation3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>rolex oyster watch</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>sony NEX-5T</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>sony vaio</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>sony360</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>sony Kinect</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

As a result, even though we have employed a very high-dimensional visual codebook trying to enhance the discriminative ability of visual representation, Table 1 has also demonstrated this point. For example, in the annotation of Xbox Kinect, there are plenty of Xbox360 found, which thus degrades the average precision. It may become even worse if we add more products that are usually close to each other. Therefore, it is important to develop methods to further enhance the visual signatures. The use of descriptive visual features (such as [Chum et al. 2007][Philbin et al. 2007]) and robust logo detection (such as [Gao et al. 2009][Kleban et al. 2008]) can be a choice.

Here we also emphasize that our approach is computationally efficient. After feature extraction, the annotation of a product for a frame actually scales as $O(M)$, where $M$ is the number of non-zero bins of the visual signature. Table II shows the average number of non-zero bins of the 20 visual signatures obtained by different methods. From the table we can see that the sparsification methods
will dramatically reduce the number of non-zero bins of visual signatures. When annotating a large dataset, we can build an inverted structure by investigating the sparsity of the visual signatures. Therefore, the sparsification of visual signatures will not only improve the annotation performance but also reduce computational cost.

6.4 CONCLUSION AND FUTURE WORK

This paper presents a novel solution to automated product annotation in videos by exploring product images on the web. Given a product name, we harvested the example images on Amazon as well as Google image search engine. We collected the nearest neighbors of each Amazon example in the Google image search result set. In this way, we collected a set of positive examples and build a visual signature based on their BoVW representations. We employed a correlative sparsification algorithm to remove noisy bins in the visual signatures. These visual signatures are used to annotate video frames. A series of experiments conducted on more than 1,000 web videos demonstrated the feasibility and effectiveness of our approach. There are some future works along the direction. Our current scheme uses a higher order visual presentation of products, built based on original SIFT feature developed by David Lowe [Lowe 2004]. We plan to test more descriptive features such as those that incorporate spatial information [Jegou et al. 2008], incorporate color information [van de Sande et al. 2010][Burghouts and Geusebroek 2009], or logo recognition techniques [Romberg et al. 2011]. In addition, we will test our scheme on a larger video set, including several movies and TV programs and the semi-automatic methods [Wang and Hua 2011].

REFERENCES


ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 1, No. 1, Article 1, Publication date: January 2012.


