A GENERIC PROCESS CHAIN TO EXTRACT KEY-OBJECTS FROM VIDEO SHOTS

Jérémy Huart, Pascal Bertolino

GIPSA-lab, INPG-CNRS
Grenoble, France
Jeremy.huart@inpg.fr, pascal.bertolino@inpg.fr

ABSTRACT

This paper discusses object-based representation of video shots acquired by a moving camera. Our approach uses an extraction of foreground regions capable of representing semantic objects of interest. However, foreground regions extracted by motion compensation are not always representative of the entity they depict. A filtering and a clustering of these regions allow us to retain only the most representative of each real object in the shot, i.e. the key-object.

Index Terms— key-object, video representation

1. INTRODUCTION

The compact description of video content is currently a difficult task due to the large mass of data it contains. A classical shot representation consists in detecting key-frame(s) extracted with different features such as color, motion, . . . An overview of the major techniques for key-frames extraction can be reviewed in [1]. Recently, some researches are tending toward object-based representations [2, 3, 4, 5]. Two approaches can be considered: the first one consists in selecting, in the shot, the frames which assist the key-object extraction [2, 6]. The second approach consists in extracting, all shot long, key-regions in order to collect information about these regions and deduce a shot representation [7, 8, 4]. The method proposed can be included in the second approach.

We use a foreground region extraction method based on the irregular pyramid algorithm. The regions are extracted by motion compensation to deal with any moving camera. The segmentation process [9] is only localized on the foreground region edges supposed to match with the edges of the real object called in the sequel *Object Of Interest* (OOI). However, only a subpart of the OOI may have a detectable apparent motion between two frames. The OOI can also be partially and/or temporally occluded. Thereby, it is often impossible to extract in each frame a Video Object Plan (VOP) fully representative of the OOI. The extracted regions are often only Sub Video Object Plans (S-VOPs) not necessarily all representative of the OOI (figure 1).

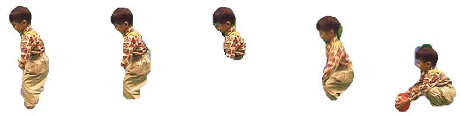


Fig. 1: Example of S-VOPs extracted from a non rigid OOI (*children* video shot) with our method

From a video first segmented into shots, we select in a generic and automatic way a set of occurrences (i.e. S-VOPs) for each OOI (*cf.* fig. 2). The most representative occurrence is called *key-object*. Therefore, we propose the following generic chain of processes for each shot:

1. Extraction of the S-VOPs
2. Rejection of low quality S-VOPs
3. Color classification of the S-VOPs : one class per S-VOP (generating S-VOP)
4. Suppression in each class of the S-VOPs that are not spatio-temporally coherent with the generating S-VOP
5. Fusion of the classes to provide one class per OOI
6. Rejection of the classes temporally not relevant
7. Selection of the key-object for each class

Each stage of the process can be viewed as a black box provided with a finite number of inputs/outputs. In this way, any of these boxes can be eventually replaced by another one, more efficient or dedicated to a particular application.

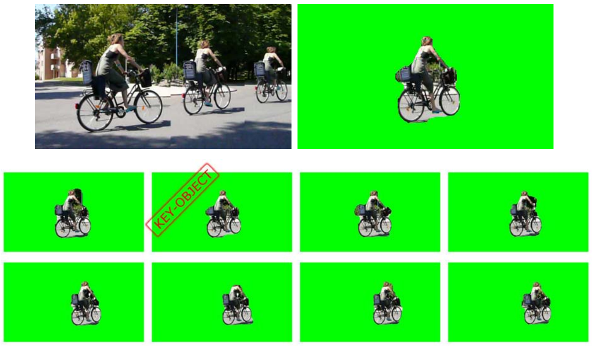


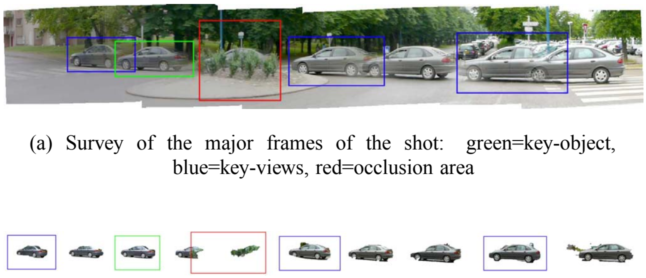
Fig. 2: Example of processing using our method. Top left: montage with 3 frames to give an overview of the shot. Top right: the key-object extracted. Bottom: some S-VOPs extracted

1-4244-1437-7/07/$20.00 ©2007 IEEE

2. APPLICATIONS

Here, we enumerate a couple of applications that could benefit from the key-objects extracted with our chain:

1. The intra-shot selection of the key-objects associated to an inter-shots object clustering method [10] can provide the entire video object-based description.
2. Any key-object can provide an automatic initialization of a tracking method based on partition projection [11]. Other S-VOPs (key-views) can be used to control the quality of the tracking throughout the shot here and there and to solve occlusion problems (fig. 3).
3. Object-based indexing and retrieval.



(a) Survey of the major frames of the shot: green=key-object, blue=key-views, red=occlusion area

(b) Some results of the tracking obtained

Fig. 3: Tracking controlled by a key-object and key-views

3. THE EXTRACTION CHAIN

3.1. Extraction of the set of S-VOPs

The extraction of the S-VOPs can be achieved using any technique that provides for each frame a set of masks of moving entities. We used a fast technique that computes a parametric global motion model for each frame [12] coupled with an accurate segmentation of the moving region borders [9]. This step provides a set of masks (S-VOPs) for the whole shot, each one attached to a frame number.

3.2. Rejection of low quality S-VOPs

Any S-VOP not representative of an OOI must be removed from the S-VOP set. The rejection is based on 2 criteria: the first one makes the assumption that a good S-VOP has a compact shape while the second one requires a good matching between the S-VOP boundary and the OOI boundary, as described in this section.

3.2.1. Compactness: a discriminating feature

Any generic object extraction method induces some punctual errors in the detection and extraction of S-VOPs. One of the main drawback due to motion estimation is the "leak" from the object to the background, or conversely. A feature of such corresponding thin and/or elongated regions is a low compactness. Compactness C_1 of an S-VOP s is given by the shape factor:

$$C_1(s) = \frac{\text{Perimeter}(s)^2}{4\Pi \times Area(s)} \qquad (1)$$

$C_1 \in [1, \infty]$. In the majority of processed video shots, we have observed a principal mode for shape factor values near 1 representing compact regions. An empirical threshold fixed to 2.5 permits to exclude any S-VOP not compact enough.

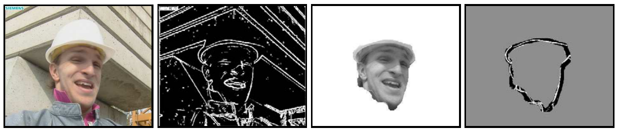
3.2.2. Edge quality evaluation

The S-VOP quality is given by the matching between its boundary and the OOI boundary. Let z be the thick outline of s and e the strong edges obtained by an adaptive thresholding of the Sobel gradient in the original frame[1]:

$$z(s) = Dilat_{\epsilon}(s) \setminus Erod_{\epsilon}(s) \qquad (2)$$

ϵ is a structuring element whose radius is only a few pixels (typically 6).

$$C_2(s) = \frac{card(e \in z(s))}{Area(z(s))} \qquad (3)$$



(a) Original frame

(b) Strong edge points

(c) S-VOP

(d) e and z (in black)

Fig. 4: Quality measure of a S-VOP

A value of $C_2(s)$ below a threshold S_2 implies the rejection of s. S_2 is shot dependent: within a shot, the distribution of the values of C_2 is modeled by a Gaussian $G(\mu, \sigma)$: S_2 must not be too restrictive in order to keep a sufficient number of S-VOPs. We chose:

$$S_2 = \mu(C_2) - \sigma(C_2) \qquad (4)$$

3.3. S-VOPs classification

At this stage, a lot of S-VOPs belong to the same OOI. This section shows how they are clustered in n classes representing the m OOI (in an ideal case, n should be equal to m). Color remains the most representative intra-shot feature of an OOI since its variations concern essentially its intensity [10]. Since m is *a priori* unknown we use a 2 stages classification:

[1] $A \setminus B = \{x \in A$ and $x \notin B\}$.

1. Each S-VOP is considered as a potential key-object and generates its own color class. In each class are gathered all the S-VOPs similar in color. Then, to integrate the spatio-temporal information, each color class is filtered to obtain trajectory-coherent classes: each S-VOP that does not verify the trajectory coherence is definitely excluded from the class.
2. Classes containing roughly the same S-VOPs are merged to get n as close as possible to m: not relevant classes (i.e. containing too few S-VOPs) are suppressed.

3.3.1. Color classification

Every S-VOP s_i generates its own color class characterized by s_i's color Gaussian mixture (classically 5 components). Every other S-VOP s_j with a similar (i.e. overlapping) mixture can join the class of s_i. Of course, the different color classes largely intersect each other.

To quantify the overlapping between two Gaussians, we use the feature proposed in [13]: two Gaussians $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$ are *c-separated* if:

$$\|\mu_1 - \mu_2\| \geqslant c\sqrt{2 \cdot max(\lambda_{max}(\Sigma_1), \lambda_{max}(\Sigma_2))} \qquad (5)$$

With $\lambda_{max}(\Sigma_1)$ and $\lambda_{max}(\Sigma_2)$ the higher singular values of covariance matrices Σ_1 and Σ_2. Two Gaussians 1-*separated* or $^1/_2$-*separated* significantly overlap. So, let m_i and m_j be 2 Gaussian mixtures modeling s_i and s_j. m_i is included in m_j if and only if each Gaussian of m_i is at the most 1-separated with one of the Gaussians of m_j. if $m_i \subset m_j$ or $m_j \subset m_i$ then s_j belongs to the class generated by s_i. The use of mixture inclusion permits not only to put in the same class similar S-VOPs. It is also a way to group subparts of a given OOI.

3.3.2. Class filtering by trajectory control

To take into account the spatio-temporal information, each S-VOP of a class is controlled to check if it has a speed vector compatible with the trajectory attached to the class: using the position and the motion compensated speed of the gravity center G_{ref} of s_{ref} a reference S-VOP, the control consists in iteratively searching the corresponding S-VOPs in the reference neighboring frames. The very first reference is the generating S-VOP s_{gen}. The search is performed in each frame of the shot in two steps: from s_{gen} to the end of the shot, then from s_{gen} to the beginning of the shot. The search is made in a circular window centered on the projection of $G_{ref} = (x, y)$ having the speed $\vec{V} = (dx, dy)$:

$$proj(G_{ref}) = (x + dx, y + dy) \qquad (6)$$

A candidate element $s_i(t)$ is temporally coherent with the trajectory of s_{ref} if and only if:

$$\|G_{S_{ref}} - G_{S_i(t)}\| \leqslant r \text{ and } \left\langle \vec{V}_{s_{ref}} \cdot \vec{V}_{s_i(t)} \right\rangle \geq 0 \qquad (7)$$

r is a research window radius relative to the radius of s_{gen}.

In equation 7 the first condition ensures the spatial coherence and the second ensures the conformity between the two S-VOPs motion directions. In a given frame, the research is processed with the following rules:

1. if no S-VOP gravity center is included in the research window, the research continues in the next frame. The window position is updated with s_{ref} speed vector.
2. if only one S-VOP verifies the conditions of eq. 7, it definitely belongs to the class and becomes the new s_{ref}.
3. if several S-VOPs verify the conditions of eq. 7, they are all kept in the class. The gravity center of the whole becomes the new G_{ref} and $\vec{V}_{s_{ref}}$ is calculated as their mean speed.
4. Every S-VOP of which the gravity center is outside the research window is excluded from the class.

3.4. Hierarchical class fusion

The color classification and the trajectory control produce many classes that contain the same elements and that concern the same OOI. In this section, we explain how these classes are merged. The class aggregation consists in building a dendrogram (i.e. a hierarchical classification) in which all the classes are iteratively merged two by two. At each iteration, only the best merge (corresponding to the best similarity in our case) is performed. Then the dendrogram is split into clusters.

The S-VOPs classes can be assimilated to sets and their similarity can be evaluated from their intersection in set theory. Let δ be the dissimilarity between two classes c_1 and c_2 verifying $|c_2| \leq |c_1|$:

$$\delta = \frac{|c_2 \setminus c_1 \cap c_2|}{|c_2|} \qquad (8)$$

$\delta = 1$ when the intersection is null. $\delta = 0$ when $c_2 \subset c_1$. Let be $c_3 = c_1 \cup c_2$. The dissimilarity between c_3 and a class c is set as follows:

$$\delta(c_3, c) = min[\delta(c_1, c), \delta(c_2, c)] \qquad (9)$$

The number of classes is then determined by a classification of the dissimilarities. The aim is to maximize the inertia between two sets E_i and F_i : let E_i be the dissimilarity set ≥ 0 and $< i$. Let F_i be the dissimilarity set $\geq i$ and < 1 (dissimilarities above or equal to 1 are excluded since they indicate totally disconnected classes). Let be $D = E_i \cup F_i$. m_D, m_{E_i}, m_{F_i} are the means of D, E_i, F_i. Then, the inertia is given by:

$$I_i = w_e d(m_{E_i}, m_D)^2 + w_f d(m_{F_i}, m_D)^2 \qquad (10)$$

Where $w_e = |E_i|, w_f = |F_i|$ and d is the Euclidian distance. The best partition is obtained with an i value that maximizes the inertia:

$$T = \underset{i}{\operatorname{argmax}}(I_i) \qquad (11)$$

3.5. Suppression of the non temporally relevant classes

In this last filtering step, we propose to exclude the classes which are not temporally relevant: a class contains S-VOPs of which the first and the last apparitions are in frames i_{begin} and i_{end}. This induces the duration (number of frames) and the persistence (apparition rate) of the class. We make the hypothesis that an OOI takes place in a shot at least during a significant duration Δ and is extracted $p\%$ of the time. Δ and p are empirically fixed and permit to validate or exclude classes. We chosed $\Delta = 50$ *i.e.* 2 seconds and $p = 20\%$.

3.6. Key-object selection

Now each class is supposed to contain several S-VOPs proper to a given OOI. One key-object can then be selected for each class c using equation 3 that estimates the segmentation quality of an S-VOP. The key-object is the S-VOP maximizing this feature in a subset $\widehat{c}$ of c. $\widehat{c}$ is obtained as follows: as criterion C_2 is a percentage, small S-VOPs are privileged face to big ones. To avoid this bias, we estimate the most representative interval of the areas of c: c is divided (by k-means) in 3 disjoint sets according to the S-VOPS areas: small, middle size and large. $\widehat{c}$ is the subset giving the highest mean quality $\overline{C_2}$. It provides the key-object.

4. EXPERIMENTAL RESULTS

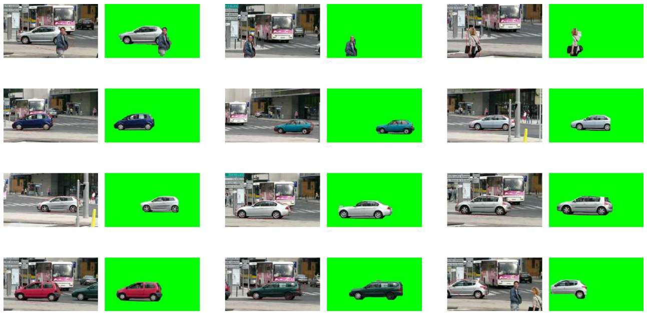


Fig. 5: The 12 key-objets extracted from the *Chavant* video. Left: the original frame. Right: the key-object

The key-objects extraction process presented in this paper can provide almost an exhaustive extraction of key-objects having a very good quality. The *Chavant* video (fig. 5) shows a crossroad filmed with the camera hand held, panning and (un)zooming. The video shot lasts 18 seconds (540 frames of size 424×240). One can clearly count 14 moving OOI. 12 key-objects were extracted. Among them, 6 cars having very similar gray colors and two pedestrians. The two missed objects were two white cars that only appeared in a small number of frames and that formed non temporally relevant classes.

The process was run on an Intel P4 2.8Ghz. The S-VOP extraction (first stage) took 20mn while the other stages took 10sec. It is to be noticed that all the temporary results are stored in image files and that the corresponding I/O are very time consuming. To conclude, let's point out that the method is neither occlusion nor zoom sensitive. The obtained key-objects can be used as robust references for many high-level video processing.

5. REFERENCES

[1] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," Technical Report HPL-2001-191, HP Laboratory, July 2001.

[2] J. Oh, J. Lee, and E. Vemuri, "An efficient technique for segmentation of key object(s) from video shots," in *ITCC '03: Proceedings of the International Conference on Information Technology: Computers and Communications*, Washington, DC, USA, 2003, p. 384, IEEE Computer Society.

[3] Ahmet Ekin, A. Murat Tekalp, and Rajiv Mehrotra, "Object-based description: From low level features to semantics," in *Proc. of SPIE Conf. on Storage and retrieval for Media Databases 2001*, San Jose, CA, January 2001, pp. 362–372.

[4] Changick Kim and Jenq-Neng Hwang, "Object-based video abstraction for video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12 (12), December 2002.

[5] H. Xu, A. A. Younis, and M. R. Kabuka, "Automatic moving object extraction for content-based applications.," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 796–812, 2004.

[6] X. Song and G. Fan, "Key-frame extraction for object-based video segmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA, March 2005.

[7] Javier Ruiz Hidalgo and Philippe Salembier, "Robust segmentation and representation of foreground key-regions in video sequences," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001.

[8] Janko Calic and Barry Thomas, "Spatial analysis in key-frame extraction using video segmentation," in *Workshop on Image Analysis for Multimedia Interactive Services*, April 2004.

[9] J. Huart, G. Foret, and P. Bertolino, "Moving object extraction with a localized pyramid," in *International Conference on Pattern Recognition*, Cambridge, UK, august 2004.

[10] R. Hammoud, *Construction et présentation des vidéos interactives*, Ph.D. thesis, Institut National Polytechnique de Grenoble, Février 2002.

[11] Guillaume Foret and Pascal Bertolino, "Label prediction and local segmentation for accurate video object tracking," in *SPIE Visual Communications and Image Processing*, Lugano, Switzerland, 8-11 July 2003.

[12] S. Liu, Z. Yan, J. Kim, and C.-C. Jay Kuo, "Global/local motion-compensated frame interpolation for low-bit-rate video," *Proceedings of SPIE*, vol. 3974, pp. 223–234, april 2000.

[13] S. Dasgupta, "Learning mixtures of gaussians," in *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1999, p. 634, IEEE Computer Society.