ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649, pp. 155–162
http://ceur-ws.org/Vol-1649, Series ISSN 1613-0073, © 2016 P. Pulc, E. Rosenzveig, M. Holeňa

Image Processing in Collaborative Open Narrative Systems

Petr Pulc[1], Eric Rosenzveig[2], Martin Holeňa[3]

[1] Faculty of information technology, Czech Technical University in Prague
Thákurova 9, 160 00 Prague
petr.pulc@fit.cvut.cz

[2] Film and TV School, Academy of Performing Arts in Prague
Smetanovo nábřeží 2, 116 65 Prague
eric.rosenzveig@famu.cz

[3] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague
martin@cs.cas.cz

Abstract: Open narrative approach enables the creators of multimedia content to create multi-stranded, navigable narrative environments. The viewer is able to navigate such space depending on author's predetermined constraints, or even browse the open narrative structure arbitrarily based on their interests. This philosophy is used with great advantage in the collaborative open narrative system NARRA. The platform creates a possibility for documentary makers, journalists, activists or other artists to link their own audiovisual material to clips of other authors and finally create a navigable space of individual multimedia pieces.

To help authors focus on building the narratives themselves, a set of automated tools have been proposed. Most obvious ones, as speech-to-text, are already incorporated in the system. However other, more complicated authoring tools, primarily focused on creating metadata for the media objects, are yet to be developed. Most complex of them involve an object description in media (with unrestricted motion, action or other features) and detection of near-duplicates of video content, which is the focus of our current interest.

In our approach, we are trying to use motion-based features and register them across the whole clip. Using Grid Cut algorithm to segment the image, we then try to select only parts of the motion picture, that are of our interest for further processing. For the selection of suitable description methods, we are developing a meta-learning approach. This will supposedly enable automatic annotation based not only on clip similarity per se, but rather on detected objects present in the shot.

1 Introduction

Amounts of multimedia content in archives of documentarists and other multimedia content creators were always large, even in the era of analogue film. With higher availability and much lower price of capturing devices suitable for cinema- or television-grade multimedia production, much more content is stored archivally and only a fraction is later published as a typical "closed narrative" ie. a traditional media work of say 30, 60 or feature length 90 minutes.

With a wider access to broadband internet connections and higher participation of individual users in the creation of internet content, the publication of such archives is now theoretically possible, yet they are usually difficult to navigate by users unfamiliar with the structure proposed by the author. Even the authors themselves tend to lose track of the entirety of their own content. And many time constrained projects or longer term project's media archives lack any structure at all.

To enable a creation of structure maintainable by a group of authors, the open narrative principle can be used. Although the original meaning refers rather to soap operas or other pieces of art with no foreseeable end, the main idea of multi-stranded narrative is easily transferable to other environments, such as documentaries.

In our example system, NARRA, that will be described in section 2, multiple strands of narrative created by multiple authors are combined and structured using data visualizations into coherent multiple narratives and can be mapped to a single graph, therefore extending the viewpoint of one author as opposed to more traditional narratives. However, such approach to multimedia clip connection discovery may be insufficient in certain cases.

One of them involves a discovery of near-identical video clips, that are created by editing the original (raw) footage. Authors tend to lose track through multiple iterated versions (including cropping, colour corrections, visual effects, retouching, soundtrack alterations or "sweetening", etc.) before arriving at a sequence used in the final edit. This brings a need for automated moving picture processing, that will be discussed in section 3.

To be able to work efficiently with only a relatively small set of interest points, instead of the whole image, common image feature extraction algorithms will be briefly presented in subsection 3.3. These algorithms will be than compared in a task of basic motion detection.

In subsection 3.4, we will present on idea of motion-based image segmentation. The basic notion is based on a similar approach used in object recognition from static images, however instead of using just the image itself for segmentation, hints from object movement will be used for object determination.

As most of the topics are still open, further research in these areas will be briefly discussed in section 4. Based on that direction of research, not only the recognition of objects, but also a recognition of the properties of the objects will be supposedly possible. In this area, we would like to use a meta-learning approach. This approach will be outlined in subsection 4.1.

2 NARRA

Open narrative systems were usually created as one-of-a-kind tools that enabled the user to browse authored content in a somewhat open manner. First approaches similar to open narrative platforms stemmed from multimedia archives at the end of 20th century, with annotations and connections curated by hand. David Blair's Waxweb, besides being the first streaming video on the web, is often cited as the first online video based navigable narrative [9].

One of the major projects of the second author, Eric Rosenzveig, on which we are building, is playListNetWork. A system developed from 2001 to 2003 in collaboration with Willy LeMaitre and other media artists and programmers. This software enabled multiple users in different locations to simultaneously work with an underlying audiovisual database, annotating the media clips and joining them into branching playlists. The publicly accessible part of the software, disPlayList, enabled a 3D visualization of the playlist structure created by playListNetWork and a subsequent unique "run" or cinematic experience through the material.

NARRA is an evolution of playListNetWork concepts, brought to a new world of hyper-linked media and direct audiovisual playback, as opposed to the more complicated multimedia streaming approaches of the past. With the increasing processing power of computers, it has been proposed that some parts of media annotation or linking can be handed over to automated processing tools.

The main task of NARRA is to create a platform for collaboration of multiple artists, and therefore the system is being built modularly, with an extensible API. During the use of NARRA on multiple projects, we discovered diverse ideas about multimedia collaboration and that different kinds of annotations are needed. To this end, NARRA uses a NoSQL database to avoid any possible limitations in the future.

Modules themselves are of three distinct types:

Connectors are used to ingest the multimedia data, yet because NARRA is not a multimedia archive, only a preview and proxy is stored alongside basic metadata.

Generators are automated tools, that process the multimedia and create a set of new metadata. An example of such a module uses an AT&T speech recognition API for automated transcription of human speech.

Synthesizers find any structure in the (meta-) data already present in storage to link the items together. For example, the synthesizer looks for a keyword similarity between two items, or is used to create and enhance links between clips used in stored video sequences.

NARRA can be then used for presentation of generated multimedia sequences, allowing for media discovery due to navigation during sequence playback or to show any user interface or visualization created in Processing.js or P5.js scripts.

This article will propose a generator creating annotations based on motion vectors in the video. Further research is intended to create a synthesizer that will enable a final linking of similar audiovisual clips automatically.

Detection and description of objects is proposed as another metadata generator. Currently, motion vectors can be used for detection of individual objects in unconstrained motion picture. Evolving rules connecting the detected objects with salient features contained in their description is a goal for our further research.

3 Moving Picture Processing

Computer vision, moving picture processing and still image processing are interconnected areas that use a very similar set of processing techniques. Using edge detection to create outlines of objects in the scene, detecting occurrences of previously defined shapes, detection of interest points and registering them among multiple pictures, etc.

Opposed to static image, moving picture brings a possibility of motion detection, yet on the other hand a problem of high data amounts that we need to deal with.

3.1 State of the Art

Many of the traditional approaches analyse individual multimedia frames, and such extracted data is taken as a discrete time sequence. Or even only statistical properties of such sequence are used for further processing.

Examples of single-frame processing methods include the classification of textures [14], bag-of-features classification [12], text recognition [11], object recognition [2] or face recognition [16].

The method created by Lukáš Neumann and Jiří Matas [11] has been also further extended into text transcription from live video. But opposed to a later mentioned approach of Fragoso et al. [3], frames were still processed one-by-one.

Other systems process pairs of frames, but have to introduce certain limitations to the acquisition process – such as limiting the motion of either the camera or the object. The camera motion limitation is for example acceptable in security camera applications, the second one in static object or environment scanning.

Especially the static camera is widely used, as it allows us a very simple motion detection concept: If many pixels change significantly in-between frames, it can be assumed that motion had happened. The location of the changed pixels tells us the position of such a motion and the difference between positions in individual frames can be deduced as a motion vector.

If we have enough information about the background or gather it during the processing, it can be subtracted from all frames to enable not only the detection of movement, but detection of whole objects. Yet still, the camera has to be static and the gathered background has to be as invariant as possible, which is not always achievable.

To enhance the information from image segmentation, other specialised sensors or camera arrays can be used to gather a depth information, however distance sensors do not usually have high-enough resolution and scene reconstruction from multiple sources is costly. There is a new method developed by Disney Research Zurich [5] to eliminate such problems, yet they are still based on processing of individual pixels into 3D point clouds.

Another problem that is currently based mostly on still image comparison, is measuring similarity between individual clips. Existing approaches try to gather similar patches from two sets of frames and compare them with invariance to very little or no editing operations [15].

3.2 Interest Point Based Image Processing

A very different approach to image processing can be based on detection and registration of interest points among a set of individual multimedia frames. This brings an advantage of much smaller data processing requirements with only a slight compromise in quality and precision. Technically, the worst type of error is a detection of similar, yet not related, points of interest. But these outliers can be filtered out later on.

To contrast with previously mentioned methods, we try to use primarily the information about interest points, especially their motion. An example of such use of motion tracking can be seen in the already mentioned translation application [3]. Image is sent to the recognition service only once, and the returned result is kept in track with the moving picture thanks to extracted motion vectors.

Image segmentation, as another example of widely used image processing technique, have to be still based on the image information itself, yet the motion information can be used to discover and track position of the detected object.

In our use case, the motion vectors extracted from all frames can be divided into two basic groups – motion of the camera itself and motion of the objects in the scene. For both groups, we can make some basic assumptions that will help us to distinguish them. In case of object motion, we can safely assume that the singular motion vectors exceeding some interframe distance are false detections and can be avoided. Also, we can assume that the motion of the object is at least to some extent smooth. Therefore, rapid movement of an object is impossible without a jump-cut in the post-production. And higher frame rate footage will be supposedly able to rely on this property even more.

The camera motion can be proposed as a smallest deviation to a global motion model. However, several problems arise as the camera can not only translate and rotate, but also change focus and in case of some lenses also zoom. The detection model therefore needs to incorporate all possible deformations of the field.

Currently, we will incorporate such moving picture description into NARRA, as a more robust computation of item similarity. By combining this approach with meta-learned rules concerning item description, we should be then able to correctly describe both the environment where the action takes place and the objects themselves. However, to validate the applicability of such a complex description, more experimentation with extracted image features and segmentation needs to be performed.

3.3 Experiments with Image Descriptor Matching

Because of distinct properties of currently used image feature descriptors and specificity of our use-case, we used the following image descriptors with two distinct matching algorithms. Brute-force (BF) searches for the closest descriptors directly, in linear time, and ends. More elaborate Fast Library for Approximate Nearest Neighbours (FLANN) [10] first creates a set of binary trees and indexes all descriptors. During search, the trees are recursively traversed many times to increase match precision – currently 50 times, which is possibly excessive. In both cases we perform a ratio check proposed by Lowe [7].

Scale-invariant feature transform (SIFT) is an algorithm for detection and description of local features in images, published by David Lowe in 1999 [7]. This algorithm takes the input image, and returns a description of the individual interest points as 8-binned gradient direction histogram of 16×16 surrounding blocks, collected on 4×4 sub-blocks. Therefore, SIFT creates a vector of 128 numbers for each interest point.

Speeded up Robust Features (SURF) is merely an enhancement of the SIFT descriptor. The Laplacian of Gaussian used in SIFT is approximated with a Box filter, and both orientation assignment and feature description are gathered from wavelet responses. Around the interest points, 4×4 sub-regions are considered, each being described by four properties of the wavelet responses. SURF descriptor therefore creates by default a vector of 64 values for each interest point.

ORB is a fairly new image feature descriptor presented by Rublee in 2011 [13] which uses Binary Robust Independent Elementary Features descriptor of detected points of interest.

Due to the binary nature of ORB, the search of matching points of interest is much faster in case of either algorithm, as can be seen in Table 2. Yet the resulting set of matches

Table 1: Number of detected motion vectors in a 50 frames long clip, frame 29 is shown in Figure 1

Resolution	ORB BF	ORB FLANN	SIFT BF	SIFT FLANN	SURF BF	SURF FLANN
480 × 270	16 232	18 200	23 830	23 554	35 628	34 312
960 × 540	15 290	16 255	49 937	49 249	120 790	116 561
1920 × 1080	14 321	14 102	165 376	161 337	387 938	363 926
3840 × 2160	13 740	12 926	1 590 839	1 496 823	1 007 805	835 092

Table 2: Computation time [s] needed for motion vector detection in a 50 frames long clip, frame 29 is shown in Figure 1

Resolution	ORB BF	ORB FLANN	SIFT BF	SIFT FLANN	SURF BF	SURF FLANN
480 × 270	7.600	7.872	27.576	30.268	51.411	51.608
960 × 540	14.200	13.796	84.212	97.040	243.257	284.860
1920 × 1080	39.280	39.248	476.160	702.644	1 601.080	2 094.176
3840 × 2160	127.700	128.960	26 868.699	57 399.776	8 997.271	12 667.774

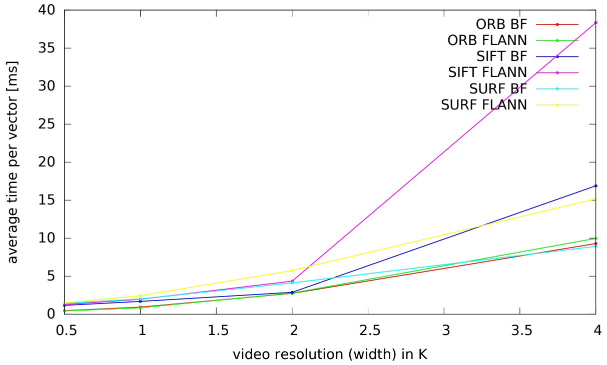


Figure 2: Comparison of time requirements for computation of one motion vector.

consist of much fewer points. The number of resulting vectors is shown in Table 1

Visual comparison of detected motion by all three algorithms is shown in Figure 1. It indicates that ORB would be useful for direct classification of actions in the image and possibly also the multimedia clip comparison. Whereas SURF, as a most time-consuming method with results exceeding the ones of SIFT by much more detected motion vectors, would be beneficial for the detailed image segmentation. However, as we are working on a proof of concept only, the much faster ORB descriptors will be in focus of our further interest.

The graph in Figure 2 also shows that SIFT does not scale well and that simple brute force based matching has a better time performance. Yet the visual comparison of outputs in Figure 1 shows that vectors matched by the FLANN algorithm are more precise. Meaning that not as many false motion vectors (long green lines) are detected. Also, the FLANN algorithm can be tuned a lot, for example by reducing the number of checks. Therefore, image segmentation will be tested on vectors obtained from ORB descriptors matched with FLANN.

It is needed to say that the current time performance of any of these algorithms is insufficient for any real-time or large archive application. Yet these results were obtained on a weak CPU (Intel 997 mobile) with no GPU acceleration, using a Python binding to OpenCV 2.4.12.2 and without any code optimization.

3.4 Image Segmentation

Image segmentation itself is a very important discipline in computer vision, as it enables to bring our focus to narrow details of a particular part of the image, as opposed to a complicated description of the whole scene.

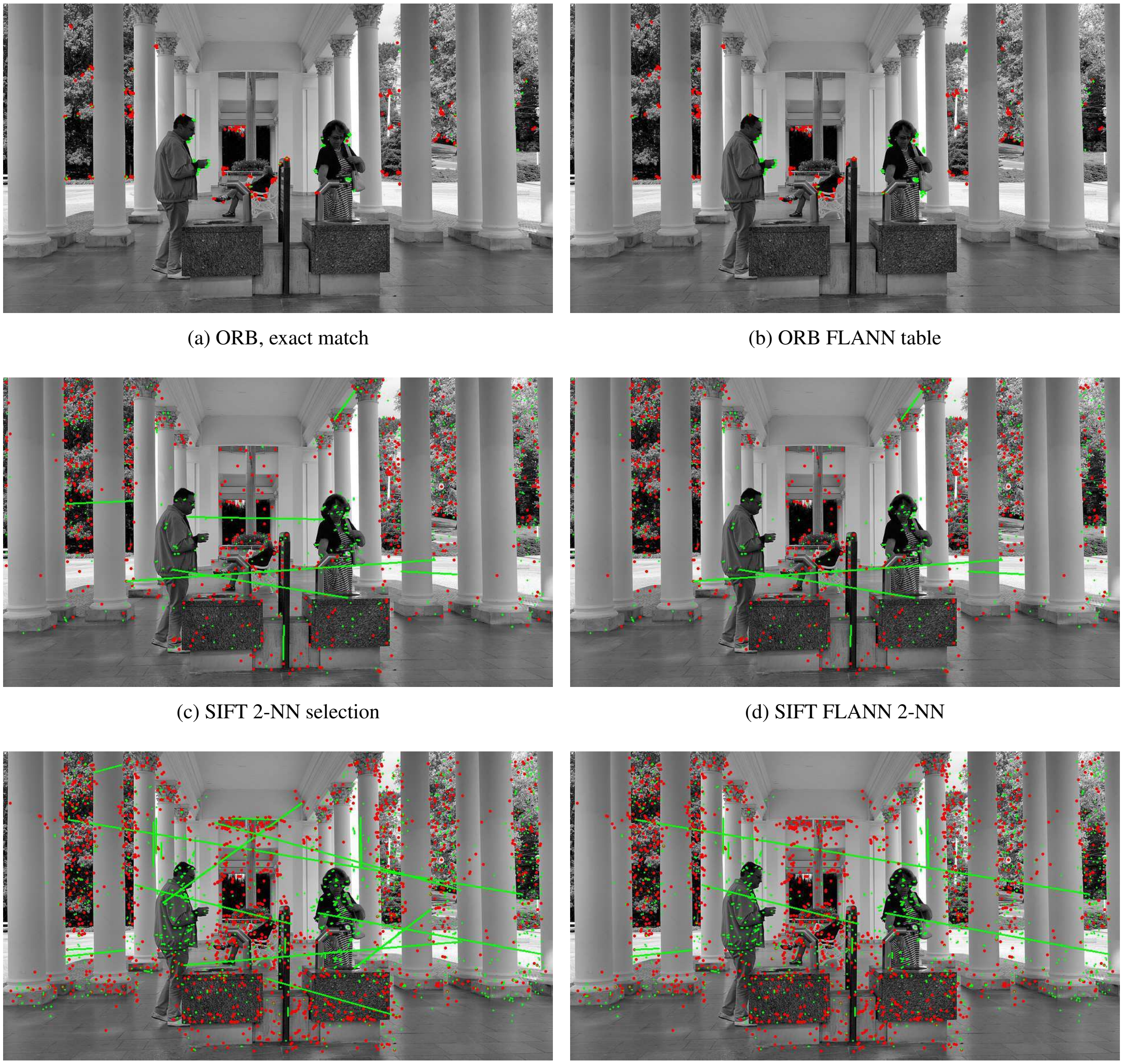
The basic image segmentation may be derived from a detection of connected components in the image and provide a set of areas, ideally affine to the local texture of the image. Such approaches, partially discussed in [8], bring a possibility to categorize such areas and therefore describe the whole image.

A bit more sophisticated image segmentation algorithm uses a principle of minimal energy cuts in the space of the image, where the inlets and outlets to the graph are assigned by rather imprecise scribbles. More precisely, we will be using a speeded-up version of Boykov-Kolmogorov algorithm – Grid Cut [4].

For better segmentation, the image is converted into an edge-representation. To this end, a convolution with the Laplacian of Gaussian kernel is performed. The inlets are then generated from the clustered motion vectors.

Such clustering is crucial as we need to assign inlets corresponding to whole objects, not individual motion vectors. To this end, all vectors of motion are represented as 6-dimensional data points, storing frame number, location and motion vector as angle sine, cosine and length.

For clustering, we have used a partially normalised data representation, where the position of the starting pixel in the image was divided by the image resolution and frame number has been made relative to the the length of the processed clip. This had a consequence that the role of those features in the performed hierarchical clustering decreased, in favour of the motion vector length and direction. Ward's linkage [17] yielded a dendrogram shown in Figure 3.



(a) ORB, exact match

(b) ORB FLANN table

(c) SIFT 2-NN selection

(d) SIFT FLANN 2-NN

(e) SURF 2-NN selection

(f) SURF FLANN 2NN

Figure 1: Visual comparison of motion vectors detected on image of resolution 960 × 540 pixels with different local image descriptors and matching algorithms. Red dots represent a vector of no motion, green lines represent a detected motion vector to next frame.

In-between shown and next frame, camera rolls slightly (at most 1 pixel distance) counter clockwise and the two people in foreground move to the right side of the image. The man on left with an average speed of 2 pixels, the woman on right with an average speed of 4 pixels per frame.

As the motion of objects and background between individual frames is minuscule, correctly detected motion vectors appear from this scale as green dots. Longer green lines are actually false matches of interest points.

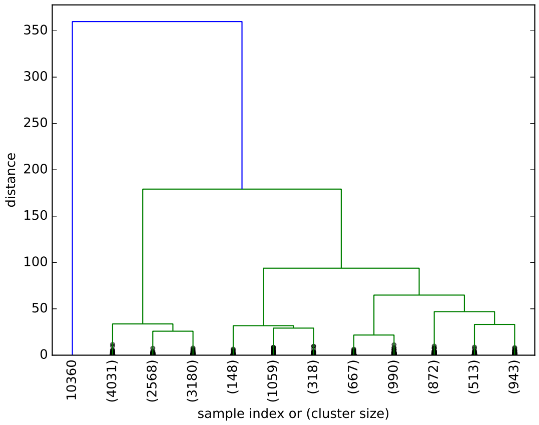


Figure 3: Dendrogram generated from hierarchical clustering of motion vectors acquired by ORB FLANN detection, see Figure 1b.

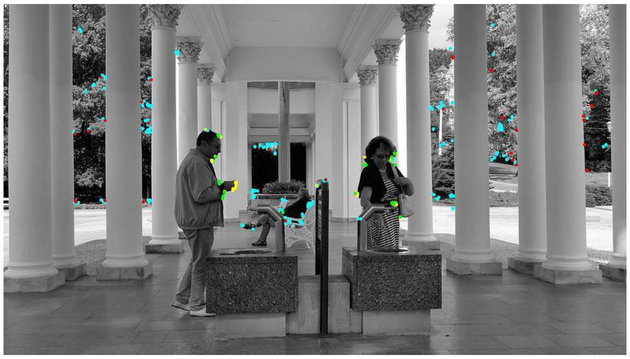


Figure 4: Resulting clustering of motion vectors.

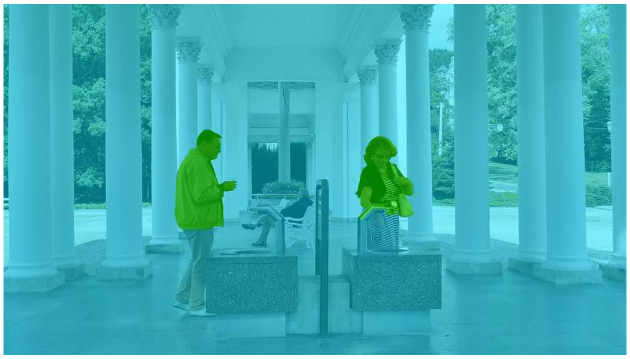


Figure 5: Resulting segmentation of the image using a Grid Cut algorithm.

Based on this dendrogram, a division of motion vectors into 5 clusters has been performed and the resulting clusters are shown in Figure 4. Cyan and red points represent correctly the background and the yellow and green points mostly represent the moving objects. Sadly, both of the objects have similar vectors of motion and normalization of point positions reduced the possibility to discriminate them.

Any detected cluster in this space is then assigned a unique descriptor that is used as a scribble index. Scribble pixels (min-cut inlets) are assigned from neighbourhood of the clustered motion vector start points.

Although we have used the approach resulting in the minimal amount of detected motion vectors, the preliminary result of segmentation in Figure 5 shows that this approach is valid and can be used at least for motion description of both the background and foreground objects. Investigating other clustering and segmentation algorithms is a part of our further research interests.

4 Further Research

The main problem with ORB descriptor is that it creates a small set of interest point matches and therefore, in Figure 5, only the upper torso containing at least some motion vectors has been correctly segmented.

Also, the process of minimal cut detection is still somewhat costly. A possible extension, that would speed-up the process significantly, is an introduction of “supernodes”, i.e., whole areas of the convoluted image will be treated as one large pixel with many connections. Although the first author had already developed such solution, its incorporation into the NARRA project will be carried out only later.

4.1 Meta-learning in Object Detection

So far, the information gathered can be used for a simple indexing tasks. For example gathering a number of objects present in the scene, their shape, colour histograms, present textures and points of interest. Motion vectors can be also indexed for later comparison of multimedia clips. Such index will be invariant to scale, crop, colour edits and other, more complex modifications of the multimedia, as the final descriptor would be deduced only from motion vectors and their relative displacement.

The final goal of our research is, however, to enable an automatic description of objects and environments in an unconstrained multimedia item. For such description, we may propose a custom baseline classifier, that would use the information about the segment contour, relative colour histogram and / or texture. However, we aim for utilisation of some already existing and previously mentioned single-frame processing methods. As the content of each multimedia segment should be now composed only of a single object in ideal case, only the classification part of such methods may be used.

Yet, we have no prior information about the type of the recognized object. The custom classifier would be difficult to train. If we would run all of the already existing classifiers and combine their outputs to deduce the final class of the object, high amount of noise and possibly contradictory information would be introduced. Also there is no sense to run the recognition algorithms on all media frames, as the ones with blurred or highly occluded objects will just confuse the classifiers.

Therefore, we are currently studying a meta-learning approaches that will select only several best-performing classification algorithms, based on the meta-features describing the considered video – such as coarsely binned colour histogram and edge information. Although meta-learning itself has been used on text corpora [1] for several decades, its application to the classification of multimedia content is rather novel.

We are currently investigating two levels, on which we can apply meta-learning to multimedia. The first, and higher-level, introduces a processing method recommendation – a classifier on the meta level that chooses the most appropriate from a set of available processing methods, based on easily extractable meta-features. In our current case, the computed boundary of segmented object, its histogram and other meta-features will be used to select more complex and thorough extraction and classification methods, such as face description or texture processing. A set of methods is used, to enable an evolution of meta-learner. To accomplish that, the best-performing method is associated with input meta-features for next rounds of the meta-learning.

This approach can be even stacked to multiple layers. An example of such situation is a more precise recognition of people, where the meta-learning classifier recognizes the shape as a human, and subsequent classification, possibly also obtained through meta-learning, brings information about recognized face, clothes, eye-wear, carried objects, types of movement and other features.

However, using more methods will also introduce much higher time complexity. To eliminate such problem, a meta-learning with multiobjective optimization can be introduced. Such meta-learning will then try to select methods both from the point of view of predictive accuracy and from the point of view of computational demands.

The second level will aim on optimization of the individual media processing units on their own. As some of the data description methods incorporate trainable and tunable methods (such as regression or classification), we can either trust their recommended settings during training, or consider multiple methods and/or their set-ups. This way, we would like to increase the precision and also possibly discover a wider variety of classes reflecting any drift in the input data.

Acknowledgements

The research reported in this paper has been carried out on Academy of Performing Arts, within project “NARRA” supported by the Institutional Support for Long-term Conceptual Development of Research Organization programme, provided by Ministry of Education, Youth and Sports, Czech Republic.

References

[1] Brazdil, P.; Giraud-Carrier, C.; Soares, C.; et al. *Metalearning*. Cognitive Technologies, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ISBN 978-3-540-73262-4. Available from: `http://link.springer.com/10.1007/978-3-540-73263-1`

[2] Duygulu, P.; Barnard, K.; Freitas, J. F. G.; et al. *Computer Vision — ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part IV*, chapter Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, ISBN 978-3-540-47979-6, pp. 97–112. Available from: `http://dx.doi.org/10.1007/3-540-47979-1_7`

[3] Fragoso, V.; Gauglitz, S.; Zamora, S.; et al. Translatar: A mobile augmented reality translator. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, IEEE, 2011, pp. 497–502.

[4] Jamriška, O.; Sýkora, D.; Hornung, A. Cache-efficient graph cuts on structured grids. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, June 2012, ISSN 1063-6919, pp. 3673–3680.

[5] Klose, F.; Wang, O.; Bazin, J.-C.; et al. Sampling based scene-space video processing. *ACM Transactions on Graphics (TOG)*, volume 34, no. 4, 2015: p. 67.

[6] Kuncheva, L. I. *Combining Pattern Classifiers*. John Wiley & Sons, Inc., 2004, ISBN 0471210781.

[7] Lowe, D. G. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, Ieee, 1999, pp. 1150–1157.

[8] Lu, W.; Li, L.; Li, J.; et al. A multimedia information fusion framework for web image categorization. *Multimedia Tools and Applications*, volume 70, no. 3, jun 2014: pp. 1453–1486, ISSN 1380-7501. Available from: `http://link.springer.com/10.1007/s11042-012-1165-2`

[9] Meyer, T.; Blair, D.; Hader, S. WAXweb: a MOO-based collaborative hypermedia system for WWW. *Computer Networks and ISDN Systems*, volume 28, no. 1, 1995: pp. 77–84.

[10] Muja, M.; Lowe, D. G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP (1)*, volume 2, 2009: pp. 331–340.

[11] Neumann, L.; Matas, J. *Computer Vision – ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III*, chapter A Method for Text Localization and Recognition in Real-World Images. Berlin, Heidelberg:

Springer Berlin Heidelberg, 2011, ISBN 978-3-642-19318-7, pp. 770–783. Available from: http://dx.doi.org/10.1007/978-3-642-19318-7_60

[12] Nowak, E.; Jurie, F.; Triggs, B. Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, Springer, 2006, pp. 490–503.

[13] Rublee, E.; Rabaud, V.; Konolige, K.; et al. ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2564–2571.

[14] Selvan, S.; Ramakrishnan, S. SVD-based modeling for image texture classification using wavelet transformation. *Image Processing, IEEE Transactions on*, volume 16, no. 11, 2007: pp. 2688–2696.

[15] Shang, L.; Yang, L.; Wang, F.; et al. Real-time Large Scale Near-duplicate Web Video Retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, New York, NY, USA: ACM, 2010, ISBN 978-1-60558-933-6, pp. 531–540. Available from: http://doi.acm.org/10.1145/1873951.1874021

[16] Turk, M.; Pentland, A. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, Jun 1991, ISSN 1063-6919, pp. 586–591.

[17] Ward Jr, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, volume 58, no. 301, 1963: pp. 236–244.