Invarivision ISS - automatic video recognition system

Dmitriy Yeremeyev
dmitriy.yeremeyev@invarivision.com

Maxim Kamensky
maxim.kamensky@invarivision.com

Alexander Krisak
alexander.krisak@invarivision.com

Abstract — We present novel video recognition system ISS that is able to find similar video fragments by matching individual frames. The basis of the technology is a template-based image recognition algorithm – AVM (Associative Video Memory), which is able to work with a huge number of images, has high reliability and high search speed. The main advantage of such approach is a capability of the system to reliably detect very small matching video fragments (starting from 4 seconds) on the big data sets. In this paper we also review the ability of ISS to recognize heavily modified video.

Keywords — template-based image recognition, automatic video recognition, associative video memory

1 Introduction

The main problem in video recognition is a huge amount of information stored inside the video stream. Systems based on digital fingerprints cope with this problem by making “fingerprints” (hash-functions that take significant features of video into account) and by rejecting the rest of information. [1,3,4,5] One can vary granularity and density of fingerprints to control quality of recognition. The main problem of this solution is that making a fingerprint requires whole sequence of frames from video, and in this case one will get strict limits on minimal length of recognized fragment.

In our case we work with separate frames of video, which are transformed to templates (recognition matrices) and placed into the search tree. To reduce the amount of stored frames we use “Frame Change Detector” that allows us very robustly to detect substantial changes in video and process only these special frames.

So this way we get storage of separate image templates with each template storing a list of markers (associations). In each marker we have an ID number of source video and position of frame inside the video.

By matching frames of scanned video with templates stored in search trees we allow system to find similar frames and localize video fragments similar to source video, previously added to the search system.

By processing separate images of scanned video, system is able to find even partial matching in cases when scanned video content was edited or suffered from moderate damage.

2 The Architecture

Invarivision Search System (**ISS**) provides distributed video processing that can simultaneously work on several servers.

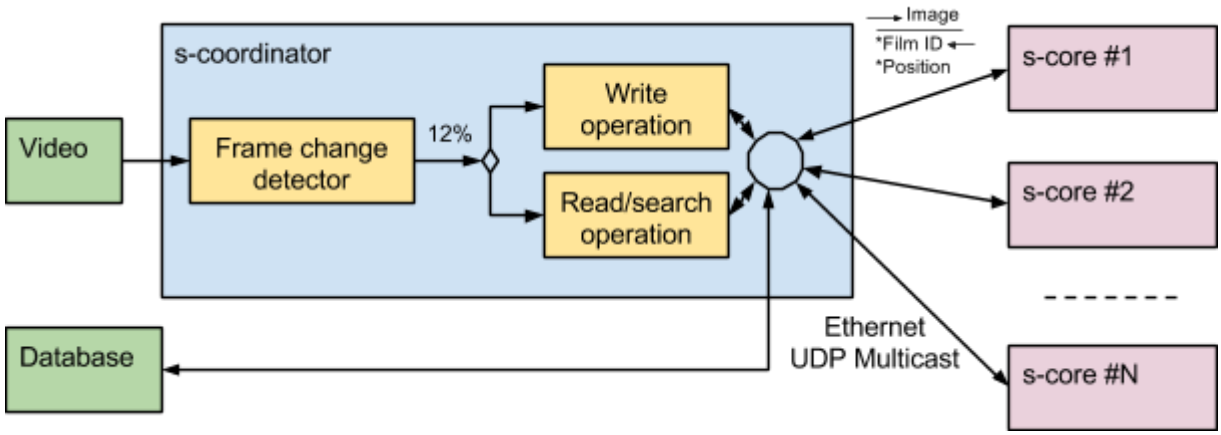
There are two types of servers:

- base server;
- node server.

The **base server** is the gateway of system that provides task management in distributed system and interacts with other node servers. This unit is self-sufficient and includes all other components of the system. Such server can work independently without additional node servers.

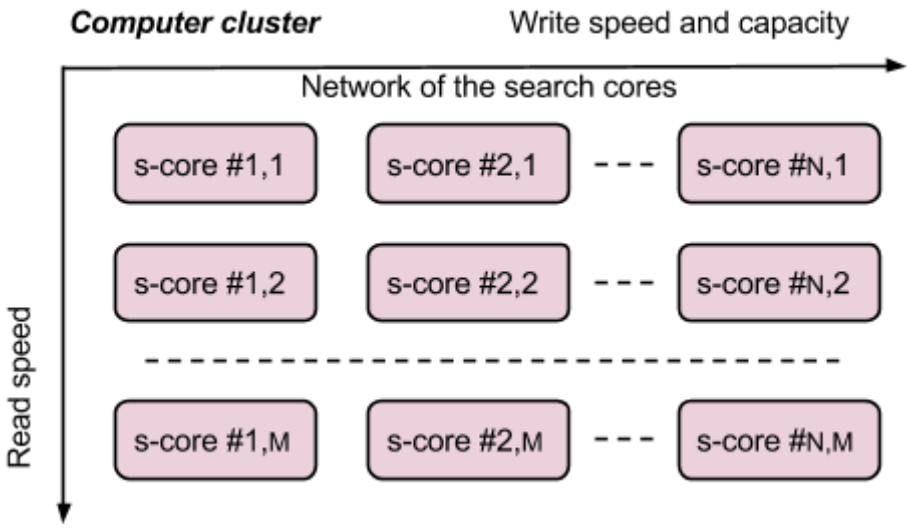
The **node server** is the unit of distributed video processing system. Such server has components for handling video recognition requests.

All these servers can contain applications for video processing and image recognition.
s-coordinator - application for coordination of video processing.
s-core - application for reading/writing of the separate images in the search tree.

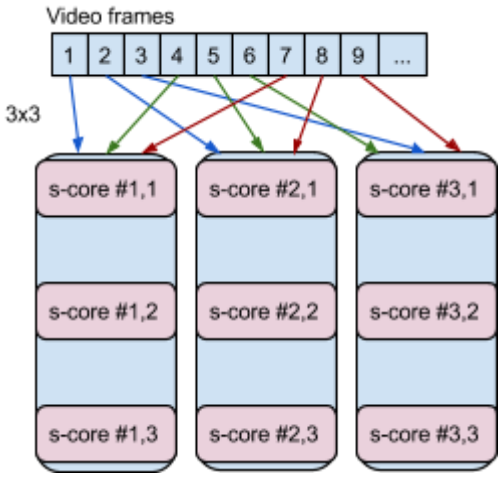


2.1 Scaling of the search system

Search system can be easily expanded with additional s-cores and node servers.
By adding s-cores to the system width we can use interleaving of frames to enhance write speed and storage capacity. And by duplicating s-core rows we can use interleaving of frames to enhance read speed.



2.2 Scheme of the video write

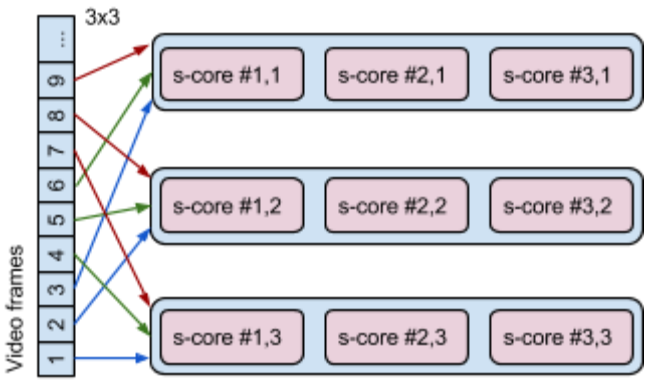


Let's examine operation of 3x3 search cluster.
The copies of rows are needed for increasing of read/search speed for each image of input video (grid computing). The each s-core rows are stored all video frames of film and each column has 1/3 of film frames in this example. All film frames were divided to the three parts and then sent to each s-core column separately for parallel computing of image write.

2.3 Scheme of the video read (searching)

Also all input video frames were divided to the three parts and then sent to each s-core rows separately for parallel image matching.

2.4 System performance



We didn't perform any extensive speed benchmarking, so we will have to do them later during another research. Latest available speed with FCD set to 12% of frames on 1 base server Dual Xeon 2xE5690 (3.47 GHz) was about 50 video hours per hour.

2.5 Storage capacity

All data of search trees are stored in RAM and it requires about 6.2Mb for storing of 1 hour of video (with FCD 12%). So, server with 256Gb RAM can store about 41290 hours of source video for searching. This value could be expanded further at the cost of performance rate by using SSD disk as a swap space. For example 1.4TB SSD drive can increase storage capacity to 225806 hours.

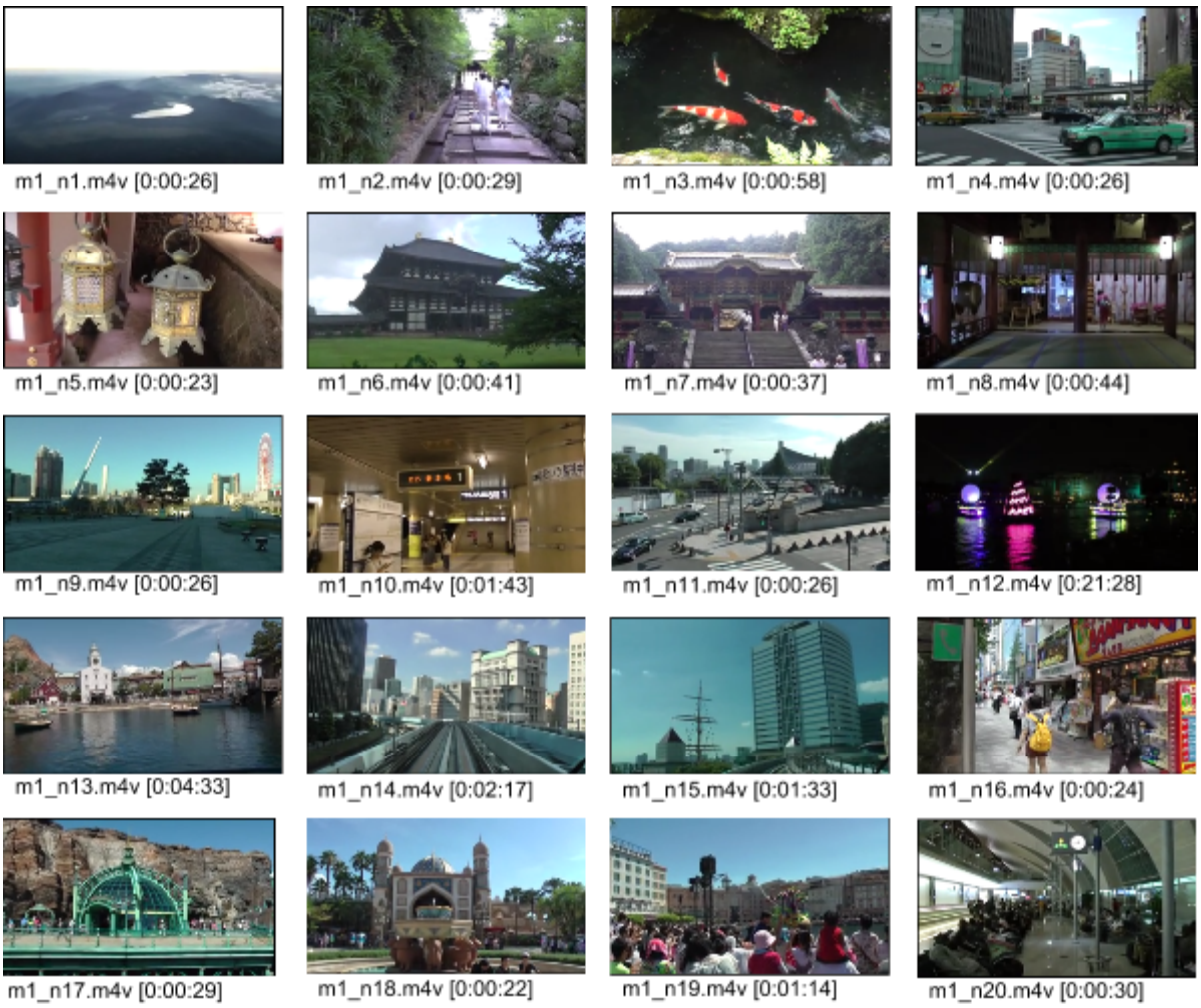
3 The Dataset

The test sets consist of 2 groups of video - “original” and “scanned”.

- original - source video added to the search system to find matches in scanned video content.
- scanned - scanned video, where we look for video fragments similar to source video.

3.1 Original set

Group “original” contains 20 videos with different scenes captured on them - people, objects, cites, night show, landscapes and so on. All original videos have 1280x720 resolutions and frame rate 50 frames/sec.



3.2 Scanned set

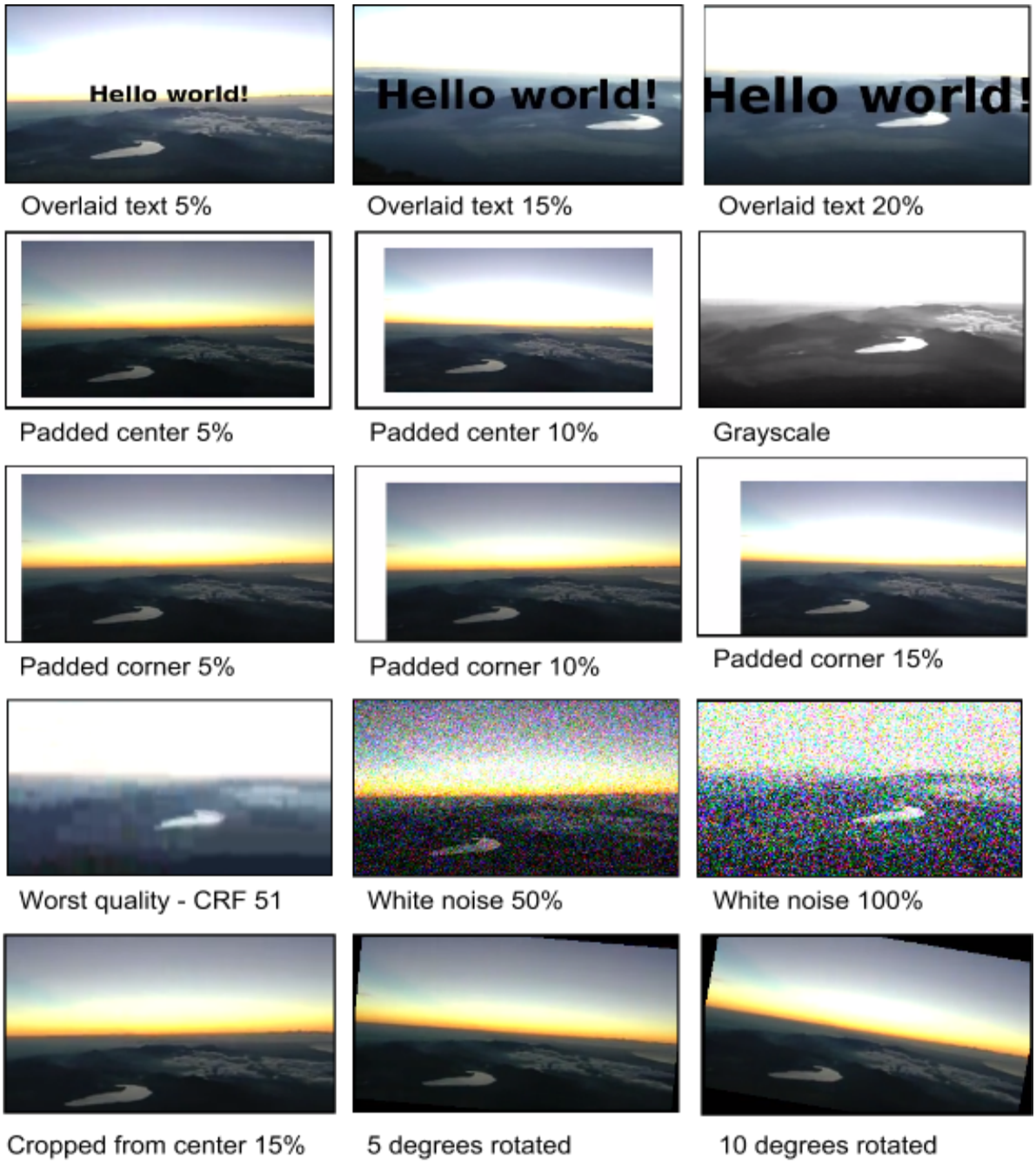
Group “scanned” contains sets of videos made from original ones. They are scaled to 25% and modified in different ways:

- rotated by 5, 7, 10 degrees
- with white noise applied - 25, 50, 100%
- padded with white color to center - 5, 10% of image width and height

- padded with white color to one of the corners - 5, 10, 15% of image width and height
- grayscaled
- re-encoded with low quality rates - 35, 43, 51 (CRF “constant rate factor” setting in x264 codec, where 0 is the best and 51 is the worst quality. Original video was encoded with CRF 20)
- with overlaid text taking 5, 10, 15, 20% of frame area.

In total we have 20 (amount of original videos) * 24 (amount of modifications) = 480 videos in scanned set. We used ffmpeg command line interface to apply modifications to our source video. [11]

3.3 Example of modifications applied to original videos



4 Testing technique

In our case we used one base server of search system (3rd release version) that was set to use 3 search cores and “Frame Change Detector” was set to select 12% of images from main videostream.

To speed up scanning and keep the storage capacity low we scaled files before applying modifications to 25% of its original size (from 1280x720 to 320x180 resolution).

All videos in "original" group must have unique content according to testing conditions. Each video in "original" group has corresponding videos in “scanned” group (one or several) with content that is similar to original.

For the purpose of calculating precision and recall values we split the timeline of each video into 1000 intervals.

Let’s say, for the sake of clarity, that:

- Original interval = interval in original video
- Scanned interval = interval in scanned video.

Then for each scanned interval we can define one of the following situations:

- True Positive (TP) — system found correct matching original interval
- False Positive (FP) — system found incorrect matching original interval
- False Negative (FN) — system didn’t find matching original interval, but it does exist

Then we can count the amount of situations occurred in scanned video - TP, FP, FN. [7,8,9,10] Having these data ready we can build Precision, Recall and F-measure for each scanned video:

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$F\text{-}measure = 2\frac{Precision \times Recall}{Precision + Recall} \quad (4.3)$$

Now we can evaluate quality measures for system and how it responds to various modifications of original videos.

5 Results

To get single quality value we calculate average of Precision, Recall and F-measure for all videos having specific type of modification in scanned video set and gather all information into table.

Table 5.1 - Test results

Data set	Average precision %	Average recall %	Average F-measure %
Original	100	98.22	99.1
25% scaled	100	99.14	99.57
5 degrees rotated*	100	96.35	98.14
7 degrees rotated*	100	51	59.01
10 degrees rotated*	100	0.71	1.4
White noise 25%*	100	98.69	99.34
White noise 50%*	100	92.06	93.5
White noise 75%*	100	93.9	96.85
White noise 100%*	100	92.34	93.65
Padded center 5%*	100	98.21	99.09
Padded center 10%*	100	36.44	53.41
Padded corner 5%*	100	97.6	98.78
Padded corner 10%*	100	88.76	94.04
Padded corner 15%*	100	17.25	29.42
Cropped from center 5%*	100	93.73	96.76
Cropped from center 10%*	100	97.28	98.62
Cropped from center 15%*	100	40.11	57.25
Constant Rate Factor 35*	100	97.46	98.71
Constant Rate Factor 43*	100	96.88	98.41
Constant Rate Factor 51*	100	93.61	96.7
Grayscale*	100	94.07	96.94
Overlaid text 5%*	100	94.18	97
Overlaid text 10%*	100	79.44	88.54
Overlaid text 15%*	100	75.17	85.82
Overlaid text 20%*	100	37.75	54.8

*first this video was scaled to 25% (from 1280x720 to 320x180 resolution)

6 Conclusion

We've got 100% accuracy on all tests due to very low False Recognition Rate of AVM algorithm used for image recognition and because of very small size of test data set.

We must notice that system has no problems with grayscale versions of videos, re-encoded video with even worst quality settings in x264 codec, with video in different resolutions.

It was a surprise for us that even 100% of white noise has almost no effect on recognition. We should try different variants of noise next time and see how system would perform in new conditions.

Search system can withstand small amounts of image rotation:

- system works without any problem up to 5 degrees
- recognition rate falls twofold at 7 degrees
- recognition is not possible at all after 10 degrees

So we can define tolerance to rotation in range [0..5] degrees. After 5 degrees we have serious degradation of recognition quality.

For the case of comparing system resistance to cropping, padding and overlaid text it's easier to evaluate these modifications by the amount of original image area left.

Table 6.1 - Test results with original image area percent

Modification	Original image area in modified image %	F-measure %
Padded center 5%	82.6	99.09
Padded center 10%	69.4	53.41
Padded corner 5%	90.7	98.78
Padded corner 10%	82.6	94.04
Padded corner 15%	75.6	29.42
Cropped from center 5%	90.2	96.76
Cropped from center 10%	81.0	98.62
Cropped from center 15%	72.2	57.25
Overlaid text 5%	95.0	97
Overlaid text 10%	90.0	88.54
Overlaid text 15%	85.0	85.82
Overlaid text 20%	80.0	54.8

We must note that during padding and cropping image becomes scaled and offset relative to original video. This complicates recognition even more because AVM algorithm normalizes image resolution.

Looking at the table 6.1 we can say that we will get acceptable recognition quality if we have 81% or more of original image area with moderate scaling and offset. With bigger modifications system stumbles upon recognition threshold and starts to reject similar frames thus dropping Recall and F-measure values.

Padding and cropping resistance can be easily increased by more extensive search (with sliding window) inside modified image. This will give us required result at cost of increased processing time.

7 Future work

System performance must be tested in controlled conditions so we can compare it later with new versions of software.

We want to perform big scale test of system for false negatives since it's very hard to detect them on already working enterprise system. This could potentially reveal some problematic cases and allow us to deal with them.

References

1. Dominic Milano; *Content Control: Digital Watermarking and Fingerprinting*
2. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton; *ImageNet Classification with Deep Convolutional Neural Networks*, 2012
3. Anindya Sarkar, Pratim Ghosh, Emily Moxley and B. S. Manjunath; *Video Fingerprinting: Features for Duplicate and Similar Video Detection and Query-based Video Retrieval*, Department of Electrical and Computer Engineering University of California, 2008
4. Wei-Lun Chao; *The Core of Video Fingerprinting: Examples of Feature Extraction*, Master's thesis, National Taiwan University, 2009
5. Xing Su, Tiejun Huang , Wen Gao; *Robust video fingerprinting based on visual attention regions*, Chinese Academy of Sciences
6. Makhoul, John; Kubala, Francis; Schwartz, Richard; and Weischedel, Ralph, 1999 *Performance measures for information extraction*, Proceedings of DARPA Broadcast News Workshop, 1999
7. *Precision and Recall* — Wikipedia
8. *Information retrieval* — Wikipedia
9. Денис Баженов; *Оценка классификатора (точность, полнота, F-мера)*.
10. Václav Hlaváč; *Classifier performance evaluation*. Czech Technical University in Prague, Faculty of Electrical Engineering Department of Cybernetics, Center for Machine Perception
11. FFmpeg Filters *Documentation*;