

Software for Detecting Irregularities in Visual Data

(No. T4-1381)

Principal investigator

Michal Irani

Faculty of Mathematics and Computer Science
Department of Computer Science and Applied Mathematics

Overview

While image & video processing are rapidly developing fields, most of the recent algorithms must be trained on large and sorted databases to gain adequate performance. Detecting irregularities in previously unknown images or videos is still an unsolved problem. Prof. Michal Irani and her team developed an algorithm that compares different segments in the same image. Only a short training is needed to detect salient areas and paint them on live. This sophisticated algorithm could be used for automated video analysis of security cameras, detecting suspicious tissue for medical applications, and for any other image/video analysis.

Background and Unmet Need

Detecting irregularities in general data, particularly in visual data, is of high importance for many applications, including detecting suspicious behaviors in video sequences, identifying salient patterns in images, and automatic visual inspection. The term "irregular" depends on the context in which the "regular" or "valid" are defined. Most systems for detecting irregularities (e.g., for surveillance or automatic visual inspection) are based on predefined rules and models. Yet, it is not realistic to expect an explicit definition of all possible valid configurations for a given context. Moreover, there are cases where an irregularity should be detected without prior examples (e.g., detecting salient regions within a given image).

The Solution

By comparing different segments in the image, the algorithm can identify which segments are irregular. In video analysis, the algorithm is more complex, and each segment is also characterized by its duration (while moving hands by walking is normal, pointing a gun for a long time will be considered suspicious- figure 1).

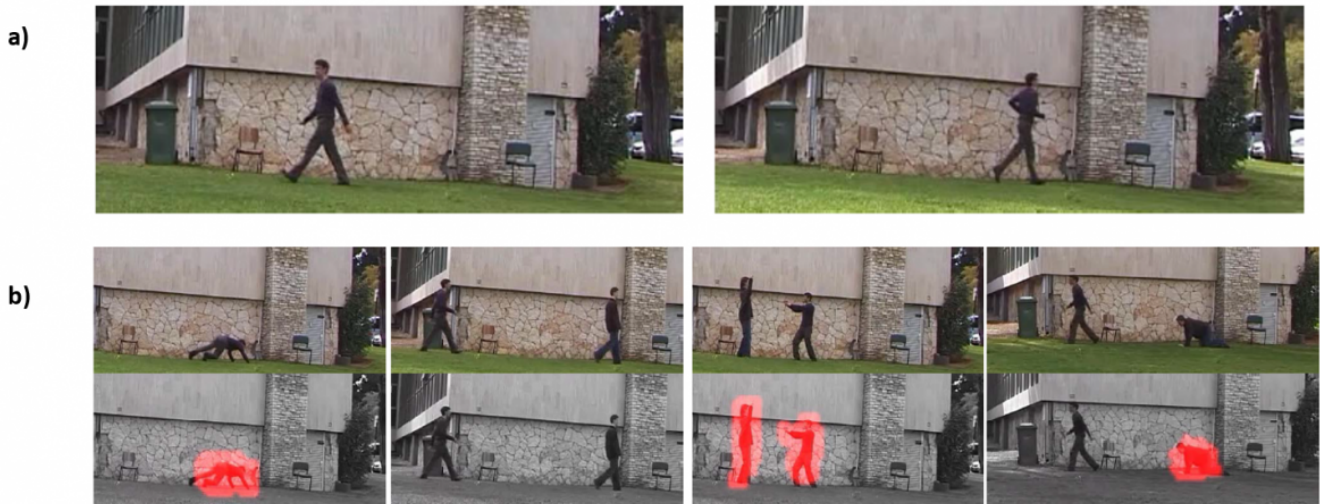


Figure 1: a) The database sequence contains a short clip of a single person walking and jogging, b) Selected frames from the query sequence: (Colored frames = input; BW frames = output; Red=Suspicious)

Â

Technology Essence

In this method, the validity of data is determined as a process of constructing a puzzle: one tries to compose a new observed image region or a new video segment ("the query") using chunks of data ("pieces of puzzle") extracted from previous visual examples ("the database"). Regions in the observed data which can be composed using large contiguous chunks of data from the database are considered very likely, whereas regions in the observed data which cannot be composed from the database (or can be composed, but only using small fragmented pieces) are regarded as unlikely/suspicious. The problem is posed as an inference process in a probabilistic graphical model. The invention also includes an efficient algorithm for detecting irregularities. Moreover, the same method can also be used for detecting irregularities/anomalies within data without any prior examples by learning the notion of regularity/irregularity directly from the query data itself.

Applications and Advantages

Applications

- Detecting suspicious and/or salient behaviors in video
- Attention and saliency in images
- Detecting irregular tissue in medical images
- Automatic visual inspection for quality assurance (e.g., detecting defects in goods)
- Generating a video summary/synopsis
- Intelligent fast forward
- Non-visual data

Â

Advantages

- No need for prior salient examples

- General to any irregularity
- Paint the suspicious segments and not the whole frame
- Fast running algorithm

Market Opportunity

Detecting irregularities in images and videos is a vital tool for automatization. It could be implied in quality assurance, especially where the type of defects cannot be predicted. It is also a powerful tool for surveillance of secure places (such as jails), where not all suspicious behaviors can be predictable, but immediate response is critical.

Patent Status

USA Granted: 8,200,648
