Enabling on-the-fly Video Shot Detection on YouTube

Thomas Steiner, Ruben Verborgh, Joaquim Gabarro,
Michael Hausenblas, Raphaël Troncy, and Rik Van de Walle

This is Carl

http://www.fantazia.org.uk/flyerlibrary/images/Stretched_Canvas.jpg

This is Victor

Carl is a canvas

http://www.fantazia.org.uk/flyerlibrary/images/Stretched_Canvas.jpg

Victor is a video

Carl and Victor are in a relationship

…and it's not complicated : -)

From the spec

To draw images onto the canvas, the `drawImage` method can be used.

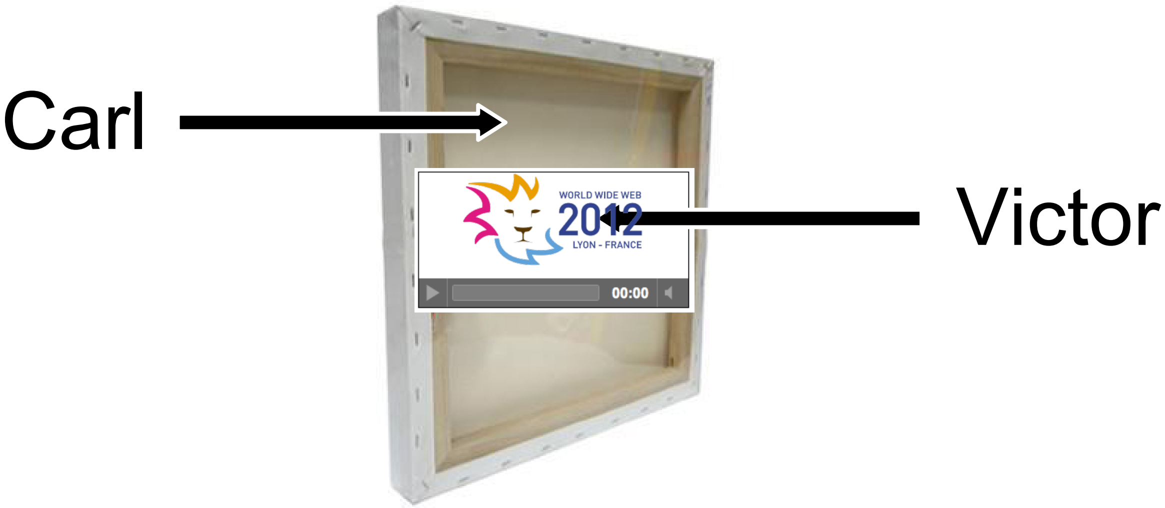
```
drawImage(image, dx, dy)
```

It can take either an `HTMLImageElement`, an `HTMLCanvasElement`, or an `HTMLVideoElement` for the image argument.

http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html#dom-context-2d-drawimage

From the spec (cont.)

When the image argument is an `HTMLVideoElement`, then the frame at the current playback position must be used as the source image



Carl

Victor

So it's party for Victor and Carl

The `getImageData(sx, sy, sw, sh)` method […] must return an ImageData object representing the underlying pixel data […].

Security with canvas elements

Information leakage can occur if scripts from one origin can access information (e.g. read pixels) from images from another origin […].

To mitigate this, canvas elements are defined to have a *origin-clean* flag.

The […] `getImageData()` methods check the flag and will throw a SecurityError exception rather than leak cross-origin data.

Shot detection

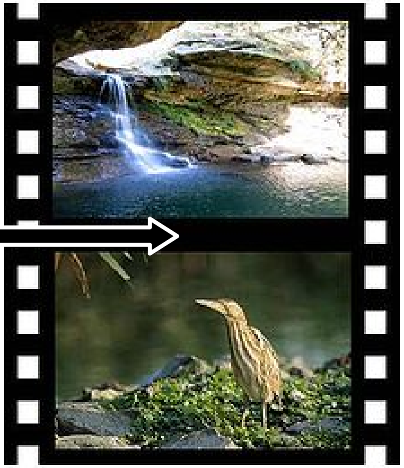
Shot detection is a field of research of video processing.

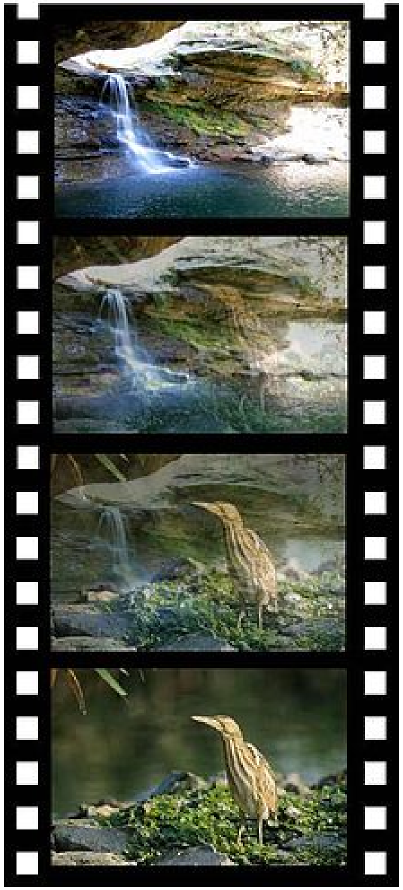
Its subject is the automated detection of transitions between shots in digital video with the purpose of temporal segmentation of videos.

http://en.wikipedia.org/wiki/Shot_transition_detection

Shot detection (cont.)

Abrupt transition





Gradual transition

http://upload.wikimedia.
org/wikipedia/commons/thumb/3/30/Hard_cut.jpg/220px-
Hard_cut.jpg

http://upload.wikimedia.
org/wikipedia/commons/thumb/b/bd/Dissolve.jpg/220px-
Dissolve.jpg

Naive approach

For each pair of frames, calculate the "distance" between those frames.

For all frame distances, if the distance is greater than a certain threshold, report a shot.

Possible distance measures

Absolute distance: sum up pixel values

+ Cheap

- Insensitive to gradual transitions

Histogram distance: compare color distribution

+ Less sensitive to fine changes

- Two frames can have similar histograms

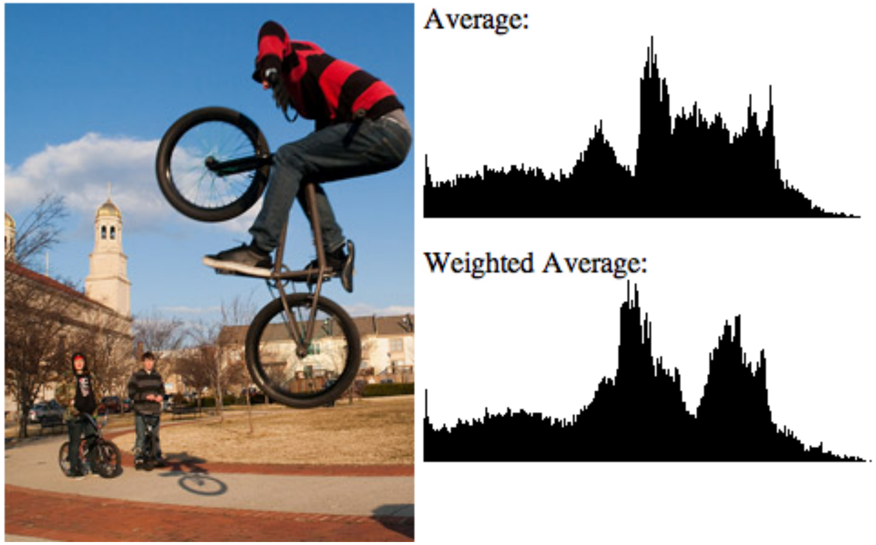
Edge detection: compare the actual content

+ High accuracy

- Expensive

Color histograms

Not all colors appear equally bright to human eyes, so the average is more heavily weighted towards green. A commonly given figure is 30% red, 59% green, and 11% blue.

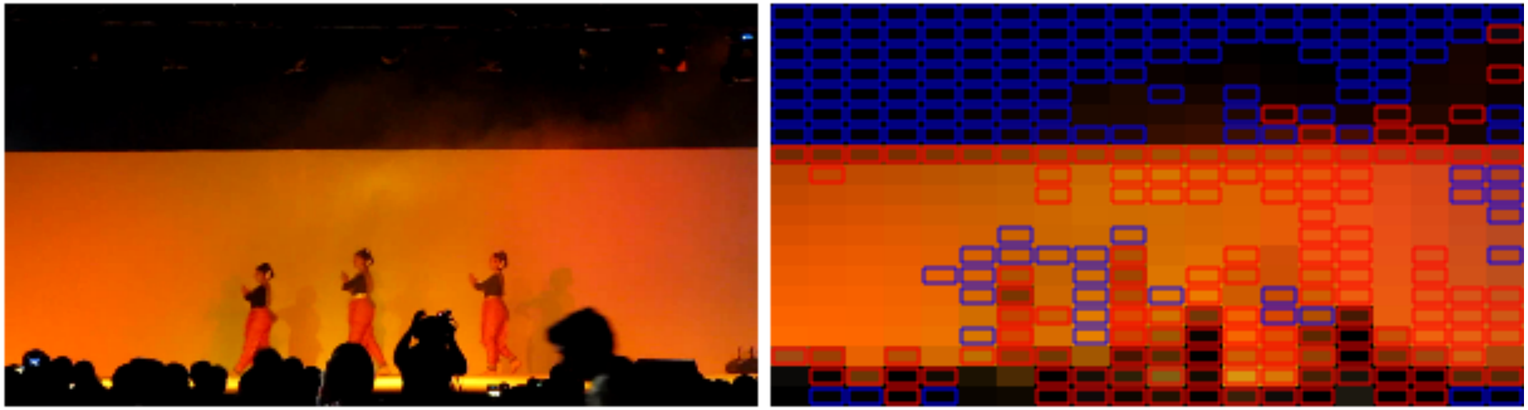


http://billmill.org/static/pixastic/demos/hist_demo.html

Dynamic tiling of frames

Idea to split video in tiles and calculate histograms tile-wise.

Can boost impact of most different tiles and limit impact of most similar tiles.



Live demo, questions

Live demo: http://tomayac.com/youpr0n/youtube.html?v=3mX4gnmvBf0

YouTube test video: www.youtube.com/watch?v=3mX4gnmvBf0

Questions: tomac@google.com or @tomayac