Timelens

und die Zukunft der Videonavigation

MRMCD 2018
@blinry
morr.cc
@blinry
An ongoing series of nonverbal algorithm assembly instructions.
HACKEN

Sudo's Priest
Cop & Bath
Children Of Eprom
Panic! At The Disco
Lamppost
ByteMish
Ramshen
#000 Sabbath
Ninjatake
#427
Byron
Ping.ME.The.Horizon
Ruby On Rails
Rasperry Pi
Ruby On Rails
Kill Nine Nails
Cult Of Vim
StudinX
The Vision
The Quiz
The Benefits
speaker cam

slide changes

free monologue
secret scene \( \theta \)
The Story
What's a good name for the video thingy on https://morr.cc/video-thingy-wip/?

- chronometric explorer
- TimeSlice

I can't decide
1163 votes on 40 ideas

Add your own idea here...
<table>
<thead>
<tr>
<th>Ideas</th>
<th>Score (0 - 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>content graph</td>
<td>79</td>
</tr>
<tr>
<td>time spectrum</td>
<td>75</td>
</tr>
<tr>
<td>visual timeline</td>
<td>71</td>
</tr>
<tr>
<td>thumbline</td>
<td>71</td>
</tr>
<tr>
<td>stripsearch</td>
<td>68</td>
</tr>
<tr>
<td>visual seekbar</td>
<td>65</td>
</tr>
<tr>
<td>thumbnav</td>
<td>64</td>
</tr>
<tr>
<td>color track</td>
<td>64</td>
</tr>
<tr>
<td>visual scrubber</td>
<td>64</td>
</tr>
<tr>
<td>visual video scrubber</td>
<td>62</td>
</tr>
</tbody>
</table>
Which name for the "visual timeline" project do you like better?

instanav

Flag as inappropriate

I can't decide

414 votes on 26 ideas

timeslicer

Flag as inappropriate

Add your own idea here...
<table>
<thead>
<tr>
<th>Ideas</th>
<th>Score (0 - 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>timelens</td>
<td>90</td>
</tr>
<tr>
<td>timescan</td>
<td>77</td>
</tr>
<tr>
<td>nordlicht</td>
<td>73</td>
</tr>
<tr>
<td>timeliner</td>
<td>72</td>
</tr>
<tr>
<td>instanav</td>
<td>69</td>
</tr>
<tr>
<td>videoslicer</td>
<td>68</td>
</tr>
<tr>
<td>timeform</td>
<td>67</td>
</tr>
<tr>
<td>aurora</td>
<td>67</td>
</tr>
<tr>
<td>timeflux</td>
<td>64</td>
</tr>
<tr>
<td>timeslicer</td>
<td>63</td>
</tr>
</tbody>
</table>
This feels like the kind of thing that we'll wonder how we lived without once we get used to it. We'll actually be able to find stuff easily in video now!

Ah, schade, dass dieser Player keinen Timelens-Support hat.

Really helpful, I want this on everything
Password1

- At least one uppercase letter
- At least one lowercase letter
- At least one digit
- At least 8 characters
The Status Quo
timelens.io
The most basic usage is to simply give Timelens the name of a video file. Timelens will create a visual timeline from the video, using a default size of 1000x100 pixels, and the default filename `INPUT_FILE.timeline.jpg`.

Run this command in a terminal:

```
$ timelens video.mp4
```

to create the following file:
You can override the filename and the timeline's size explicitly:

```
$ timelens video.mp4 --timeline timeline.jpg -w 1000 -h 500
```
Use the `--thumbnails` option to specify the name of the VTT file:

$ timelens video.mp4 --thumbnails thumbnails.vtt

For each thumbnail, the file specifies a time range and a filename, as well as the X/Y position in that file and its width and height, in the Media Fragment syntax:

```webvtt
WEBVTT

00:00.000 → 00:05.786
thumbnails-00.jpg?xywh=0,0,161,90

00:05.786 → 00:11.572
thumbnails-00.jpg?xywh=161,0,161,90

00:11.572 → 00:17.358
thumbnails-00.jpg?xywh=322,0,161,90

...
TimeLens will also create one or more JPEG files containing the thumbnails, which are referenced from the VTT file. Here's the first one:
If you're maintaining a website with video players on it, you can add Timelens support easily!

To use the web plugins, you need the `timelens.js` library, as well as the `timelens.css` file. You can install the `timelens` npm package:

```
$ npm install timelens
```

Or you can simply download them from the GitHub repository. Then, include them in the header of your website, like this:

```
<script src="/path/to/timelens.js"></script>
<link rel="stylesheet" href="/path/to/timelens.css">
```
**General-purpose JavaScript**

Suppose you have an empty div, which you want to make into a Timelens user interface:

```html
<div id="timelens"></div>
```

Use this JavaScript to specify the div’s id, and the locations of the timeline and the thumbnails VTT file:

```javascript
timelens("#timelens", {
  timeline: "/path/to/timeline.jpg",
  thumbnails: "/path/to/thumbnails.vtt"
});
```

The result will look like this:
To integrate Timelens with a video player which doesn't yet have a ready-made plugin (see below), you can use the callbacks `seek` and `position`:

```javascript
timelens("#timelens", {
    timeline: "/path/to/timeline.jpg",
    thumbnails: "/path/to/thumbnails.vtt"
    seek: function(position) {
        // This is called when the user clicks on the timeline.
        // 'position' specifies the seek position in seconds.
        your_player.seek(position);
    },
    position: function() {
        // This is called when the code wants to know the current player position.
        // to update the position of the progress marker correctly.
        // It should return the current position in seconds.
        // The progress marker will be shown only if this callback is specified.
        return your_player.position();
    }
});
```
MediaElement.js

To add Timelens support to a MediaElement.js player, add a `data-timeline` attribute to your `<video>` tag, and add a subtitle track with the label "thumbnails":

```html
<video id="mediaelement" src="/path/to/video.mp4" preload="auto" data-timeline="/path/to/timeline.jpg">
  <track kind="metadata" label="thumbnails" src="/path/to/thumbnails.vtt">
</video>
```

Also, add a "timelens" entry to the features list when initializing the player:

```javascript
var player = new MediaElementPlayer(document.querySelector("#mediaelement"), {
  features: ["playpause", "progress", "volume", "timelens"]
});
```
Clappr

To add Timelens support to a Clappr player, prepare an empty div to hold the player:

```html
<div id="clappr"></div>
```

Then add a `TimelensPlugin` entry to the plugins list when initializing the player, and define the location of the timeline and the thumbnails file in the `timelens` parameter:

```javascript
var player = new Clappr.Player({
  source: "/path/to/video.mp4",
  parentId: "#clappr",
  plugins: {
    core: [TimelensPlugin]
  },
  timelens: {
    timeline: "/path/to/timeline.jpg",
    thumbnails: "/path/to/thumbnails.vtt"
  }
});
```
The Future
Infrastructure Review and Closing

Lightning Talks

Demoshow
https://api.timelens.io/0.1/youtube/XEVlyP4_11M
gulasch
**Ready-made packages**

If you’re using **Arch Linux**, you can install the Timelens command line tool from the **AUR**, using your AUR helper of choice, e.g.

```
$ yaourt -S timelens
```

For **other platforms**, there are no packages or binaries yet (**let me know** if you want to help change that). But you can build Timelens yourself, here’s how:
Building from source

Timelens is written in the Rust programming language, so you’ll need a working Rust installation. On Unix-based systems, you’ll probably want to run these commands to install rustup and cargo (which are like pip or npm, but for Rust):

```bash
$ curl -f https://sh.rustup.rs > rust.sh
$ sh rust.sh
$ source ~/.cargo/env
```

There’s also a single dependency: The multimedia framework GStreamer. To install the required components for your platform, follow these instructions.

You can then compile Timelens like this:

```bash
$ git clone https://github.com/timelens/timelens
$ cd timelens
$ cargo build --release
```

This will create the binary `target/release/timelens`, which you can use like this:

```
$ ./target/release/timelens video.mp4
```
Interactive Navigation Summaries
Schoeffmann & Boeszoermenyi (2009)
3D Filmstrip
Hudelist/Schoeffmann/Boeszoermenyi (2013)
EXPLORABLE EXPLANATIONS
"In science, if you know what you are doing, you should not be doing it. In engineering, if you do not know what you are doing, you should not be doing it. Of course, you seldom, if ever, see either pure state."

—Richard Hamming, The Art of Doing Science and Engineering

How can we design systems when we don’t know what we’re doing?

How do we explore? If you move to a new city, you might learn the territory by walking around. Or you might peruse a map. But far more effective than either is both together — a street-level experience with higher-level guidance.

Likewise, the most powerful way to gain insight into a system is by moving between levels of abstraction. Many designers do this instinctively.
The End

morr.cc
@blinry

@timelens_io
@timelens_io