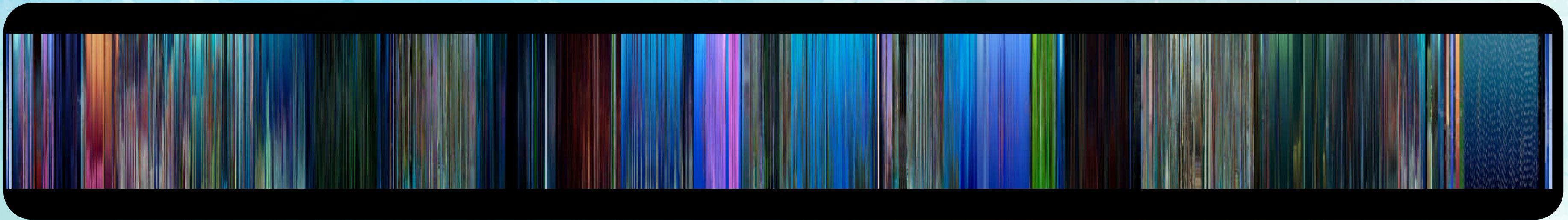




Timelens



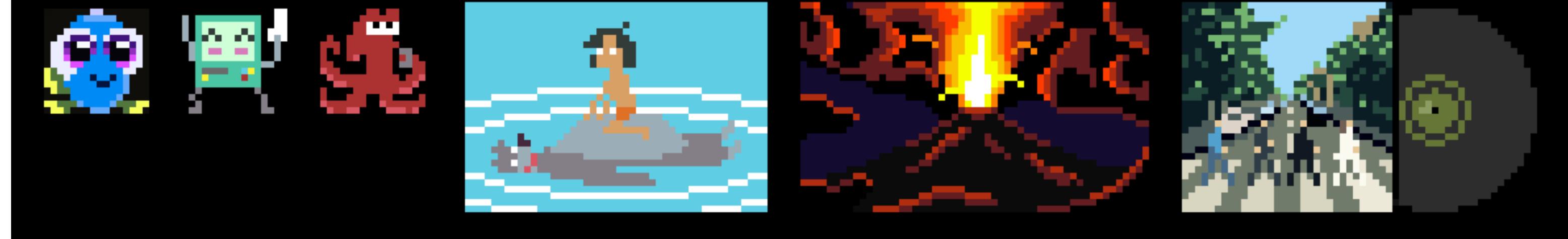
und die Zukunft der Videonavigation

MRMCD 2018

@blinry



morr.cc
@blinry





CCC Video
Operation
Center



Stratum 0



IDEA

An ongoing series of nonverbal
algorithm assembly instructions.

GIFT WRÄPPING

Based on a guest contribution by Christoph Hansknecht – v1.0, CC by-nc-sa 4.0

The diagram shows two main sections. The top section illustrates the step where a polygon is formed by connecting points. The bottom section shows the iterative steps of the algorithm: 1a (initial set of points), 1b (a point being added to a line segment), 1c (the line segment being rotated), 1d (the new point being added), 2a (the first point being marked as part of the polygon), 2b (the line segment being rotated), 2c (the new point being added), 2d (the new line segment being drawn), and 3 (the final polygon being formed).

KVICK SÖRT

idea-instructions.com/quick-sort/ v1.1, CC by-nc-sa 4.0

The diagram shows the Quick Sort algorithm with six numbered steps. Step 1 shows a random arrangement of bars. Step 2 shows a pivot bar being selected. Step 3 shows the bars being partitioned around the pivot. Step 4 shows the recursive call on the left partition. Step 5 shows the recursive call on the right partition. Step 6 shows the sorted bars and the algorithm's completion.

BOGO SÖRT

idea-instructions.com/bogo-sort/ v1.1, CC by-nc-sa 4.0

The diagram shows the Bogosort algorithm with three numbered steps. Step 1 shows a hand of cards with some being marked with a checkmark. Step 2 shows the cards being shuffled. Step 3 shows the cards being sorted into a fan-like order.

PUBLIK KEY KRYPTO

idea-instructions.com/public-key/ v1.1, CC by-nc-sa 4.0

The diagram shows the RSA public key encryption algorithm with four numbered steps. Step 1 shows a document being encrypted with a public key. Step 2 shows the ciphertext being decrypted with a private key. Step 3 shows the original document being decrypted with a private key. Step 4 shows the original document being decrypted with a public key.

HACKEN

Sudo's Priest

COPY&PASTE

Children Of EEPROM

Panic!
Kernel

POrn

ATARI
TEENAGE
ROOTKIT

ARCH
LINUX
ENEMY

LAMPP
of god Bytewish

BLINK-404

RAM STEIN INFLAMEWARS PERL JAM

#000
SABBATH NILVANA SYSTEMD OF A PWN

BASHPHEMY KEY PUNCH GREP-ZEPPELIN

TEN FINGER PING ME.THE.HORIZON SVROUTINE TO SALLY

HAMMER.CALL() MY DYING BRIDE VANS FOR MY VALENTINE

RUBY ON RAILSTORM bin EXTREMO CULT OF VIM

Raspberry Pi DEBUG MODE CANNIBAL CORES
OF FIRE

kill -9 inch nails Copyclony staticintX

The vision

moviebarcode.tumblr.com

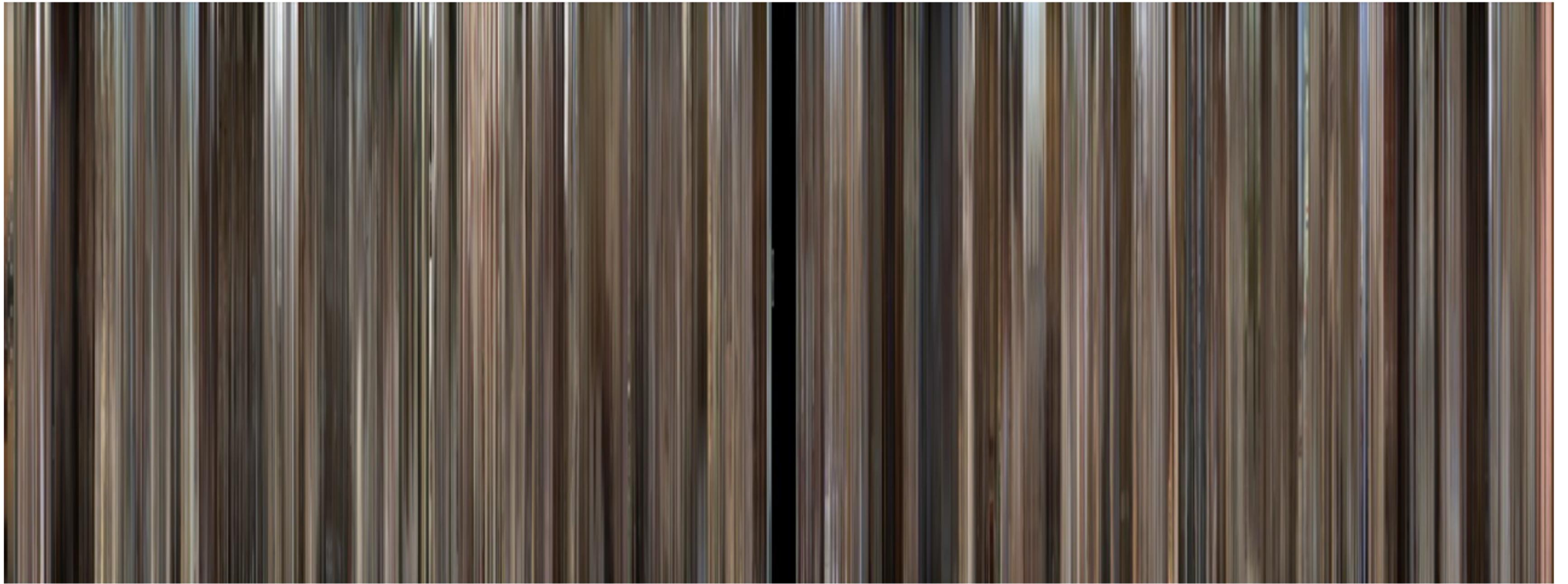
MOVIEBARCODE

[CONTACT](#) | [TWITTER](#) | [INDEX](#) | [PRINTS](#)



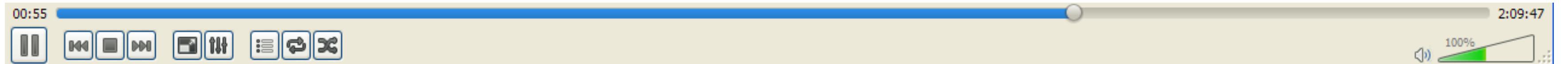
2018/05/04
147 NOTES

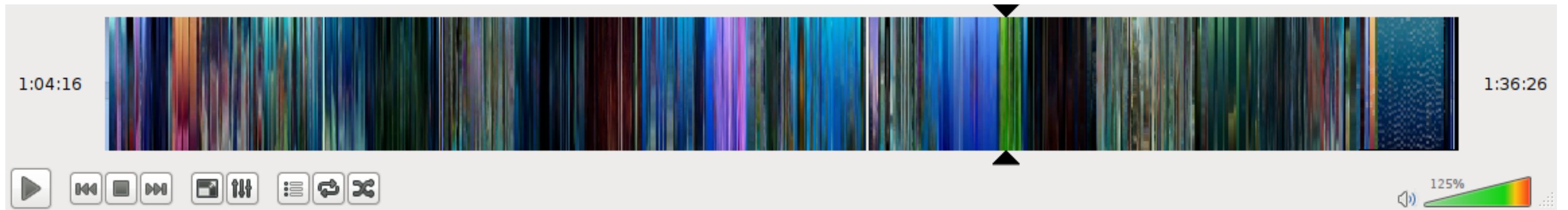
STAR WARS: EPISODE VIII - THE LAST JEDI (2017)



2012/09/03
28 NOTES

GANDHI (1982)
PRINTS





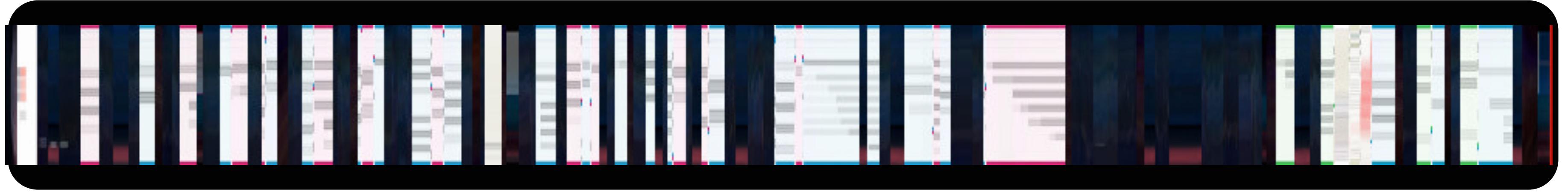
The Quiz

The Benefits



titles

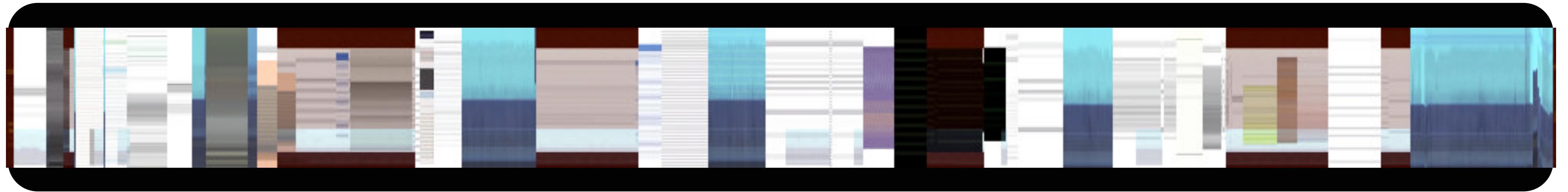
end credits



↑ ↑ ↑
speaker cam

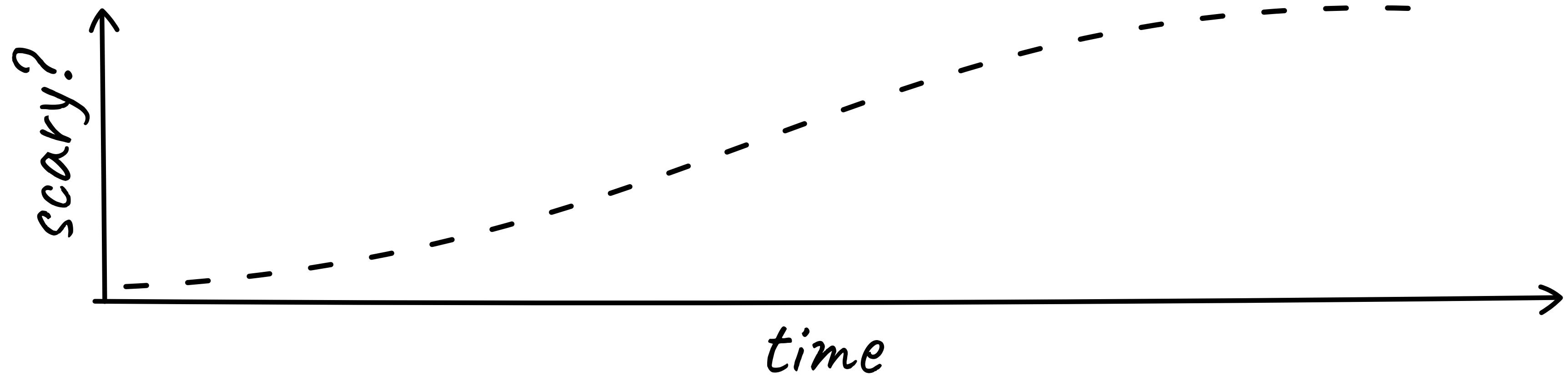
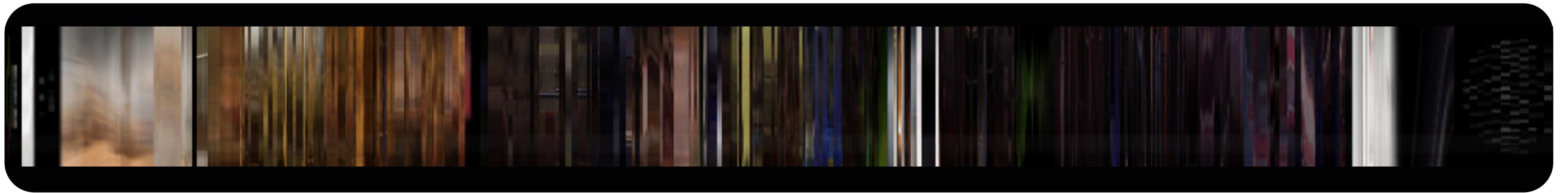
slide changes

free monologue



different sections!

Q & A





secret scene 10/



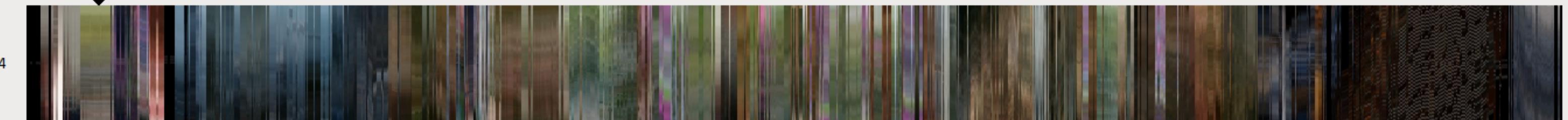
The Story

NORDLICHT



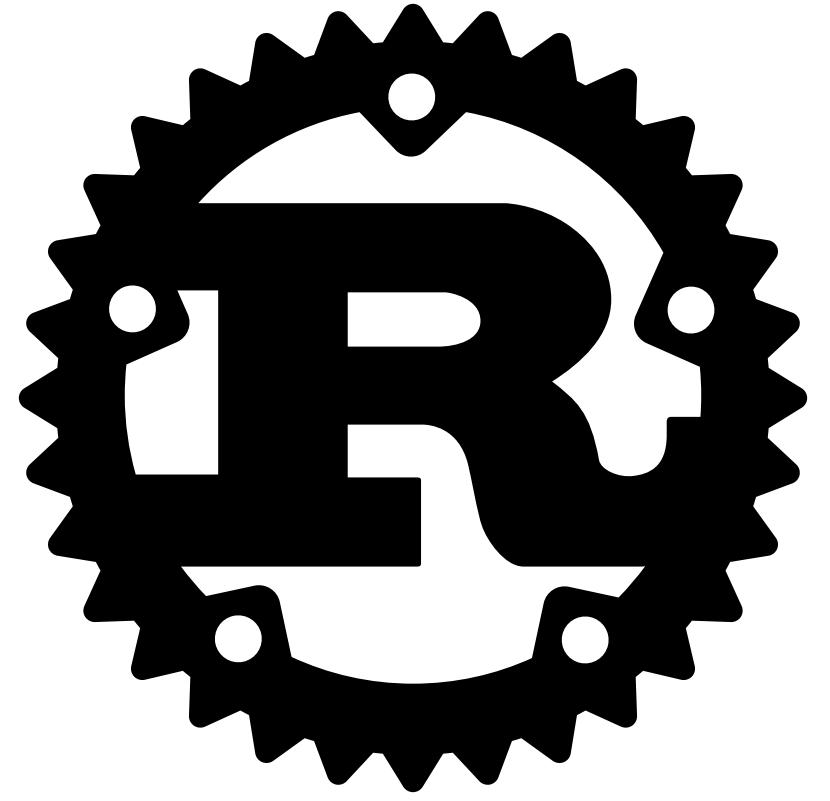


00:34



12:14





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...

Ideas

Score (0 - 100) 

content graph

 79

time spectrum

 75

visual timeline

 71

thumbline

 71

stripsearch

 68

visual seekbar

 65

thumbnav

 64

color track

 64

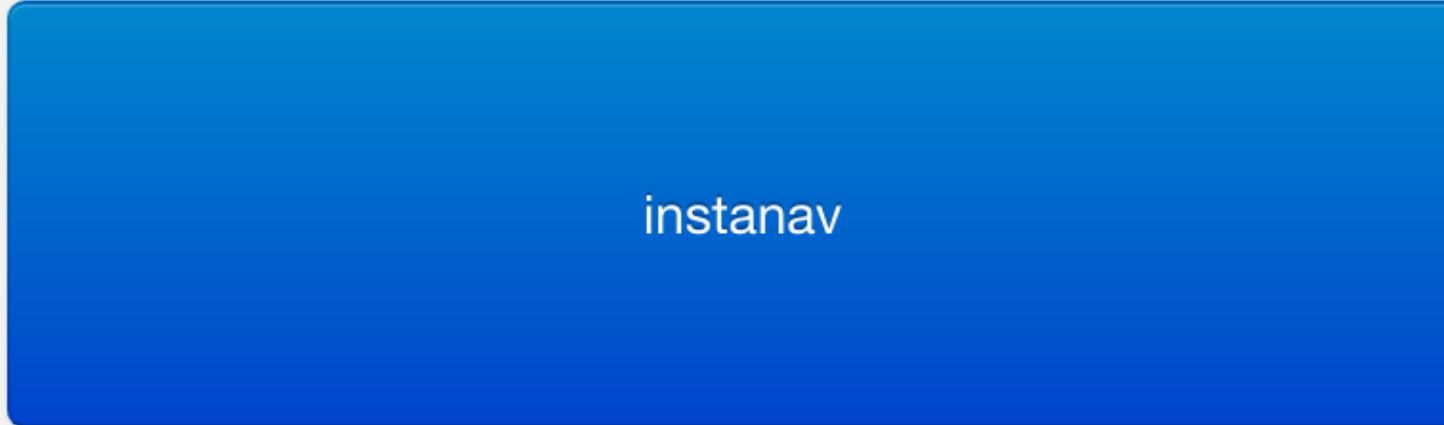
visual scrubber

 64

visual video scrubber

 62

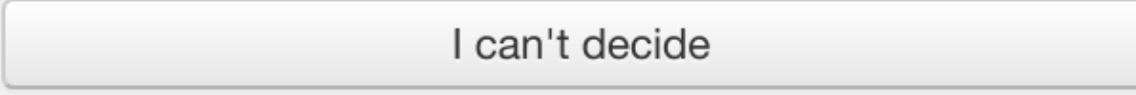
Which name for the "visual timeline" project do you like better?



instanav

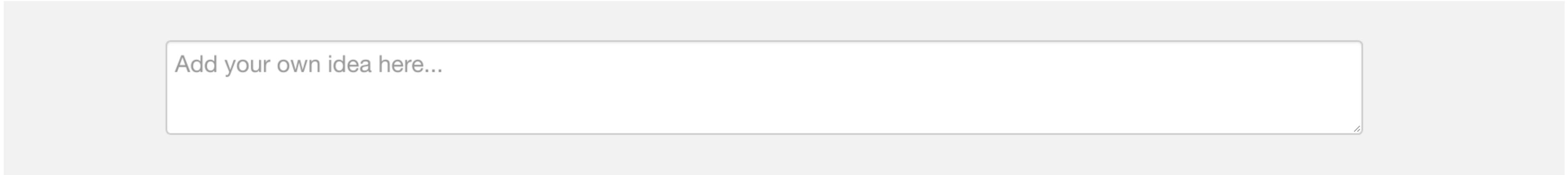
[Flag as inappropriate](#)

timeslicer

[Flag as inappropriate](#)

I can't decide

414 votes on 26 ideas



Add your own idea here...

Ideas

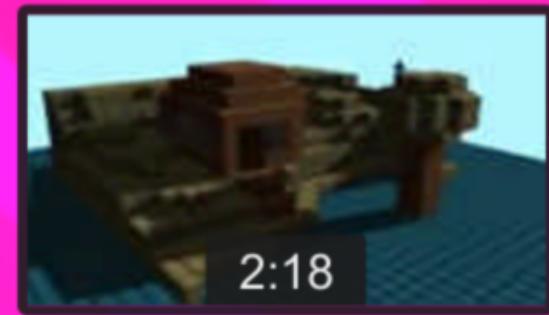
Score (0 - 100) 

timelens	<div style="width: 90%;"></div>	90
timescan	<div style="width: 77%;"></div>	77
nordlicht	<div style="width: 73%;"></div>	73
timeliner	<div style="width: 72%;"></div>	72
instanav	<div style="width: 69%;"></div>	69
videoslicer	<div style="width: 68%;"></div>	68
timeform	<div style="width: 67%;"></div>	67
aurora	<div style="width: 67%;"></div>	67
timeflux	<div style="width: 64%;"></div>	64
timeslicer	<div style="width: 63%;"></div>	63



Timelens

Ninjadev – What Are You Syncing About?



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.

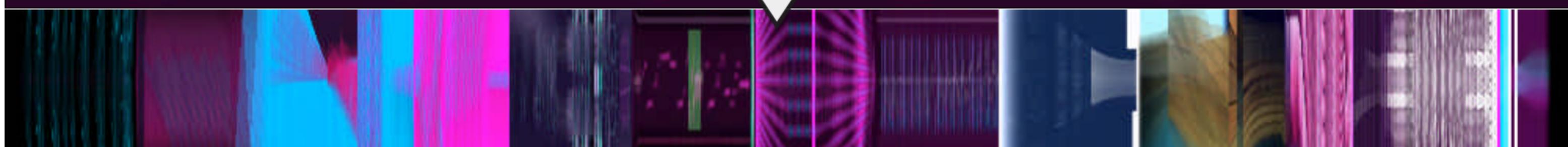
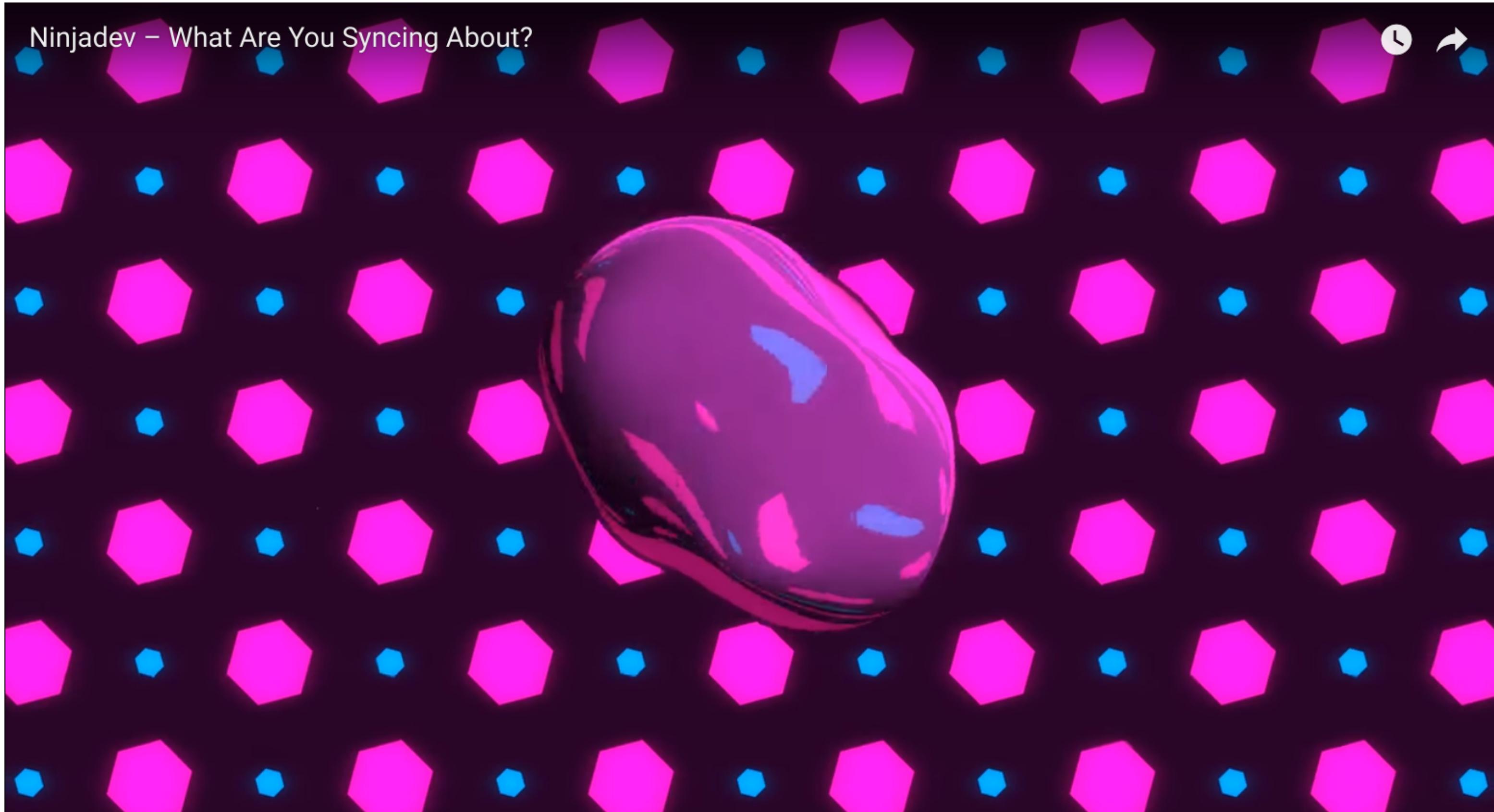
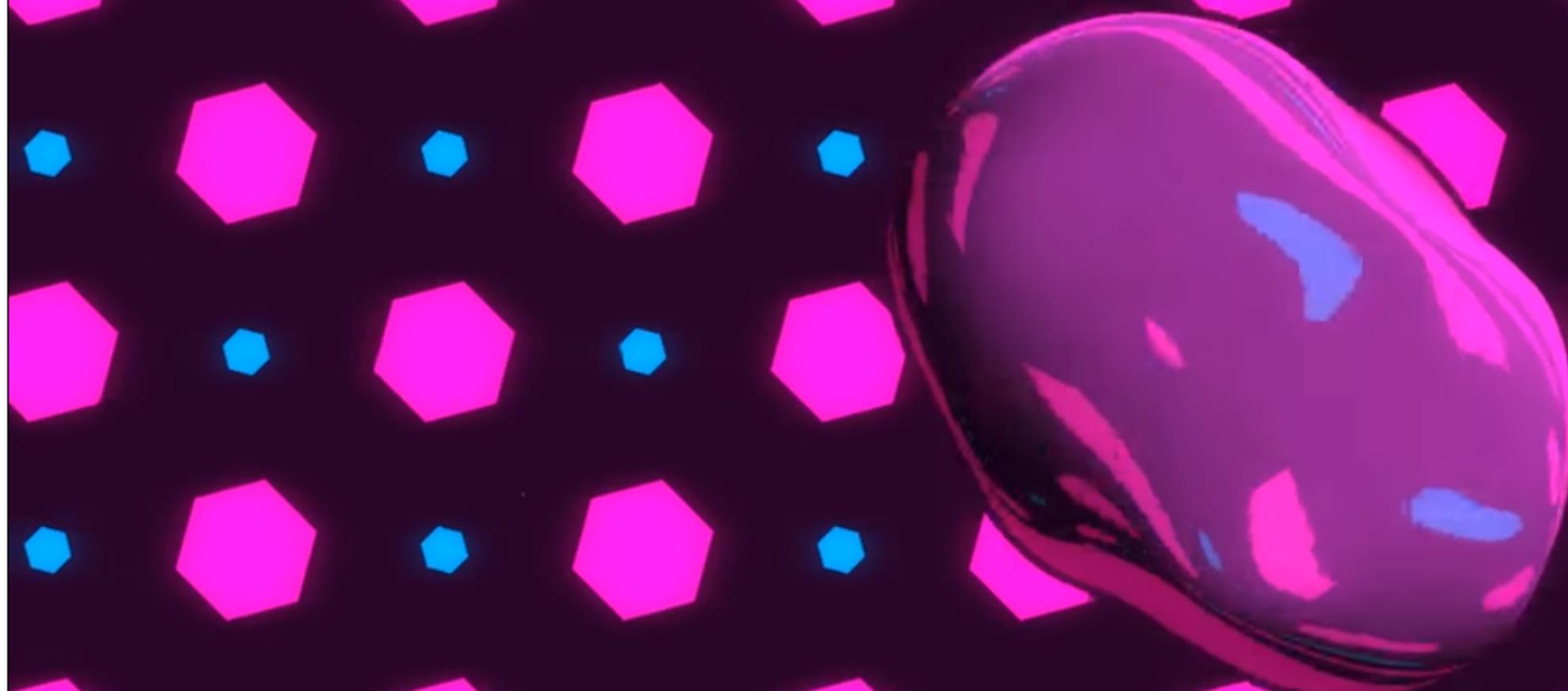
Really helpful, I want this on everything



19:51



Ninjadev – What Are You Syncing About?



Lightning Talks

pony

Password1

- ✓ At least one uppercase letter
 - ✓ At least one lowercase letter
 - ✓ At least one digit
 - ✓ At least 8 characters



The Status Quo

timelens.io



Timelens

[Home](#) [Installation](#) **Usage** [Plugins](#) [About](#)

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:

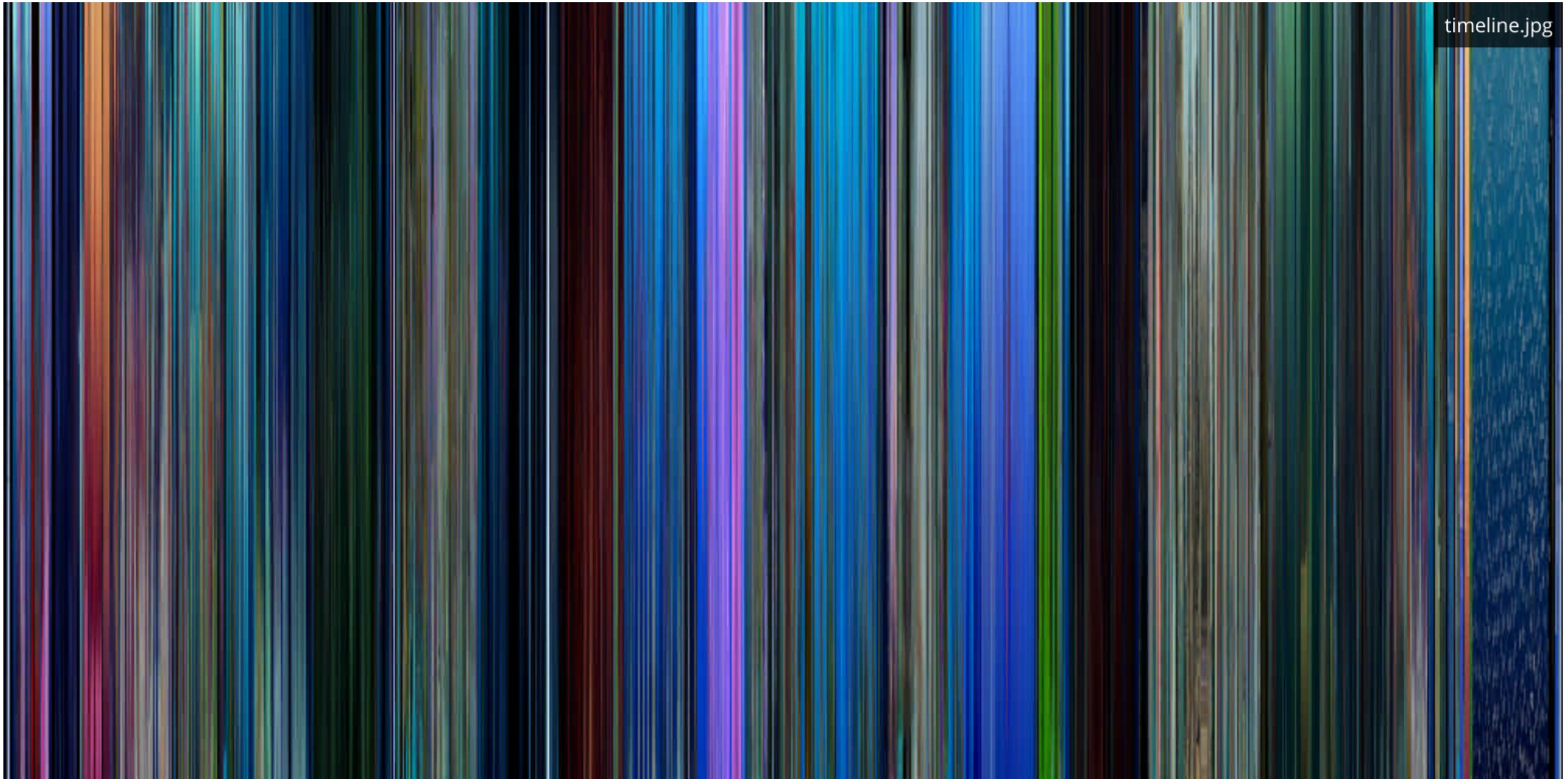
```
$ timelens video.mp4
```

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

thumbnails.vtt

00:00.000 → 00:05.786

thumbnails-00.jpg?xywh=0,0,161,90

00:05.786 → 00:11.572

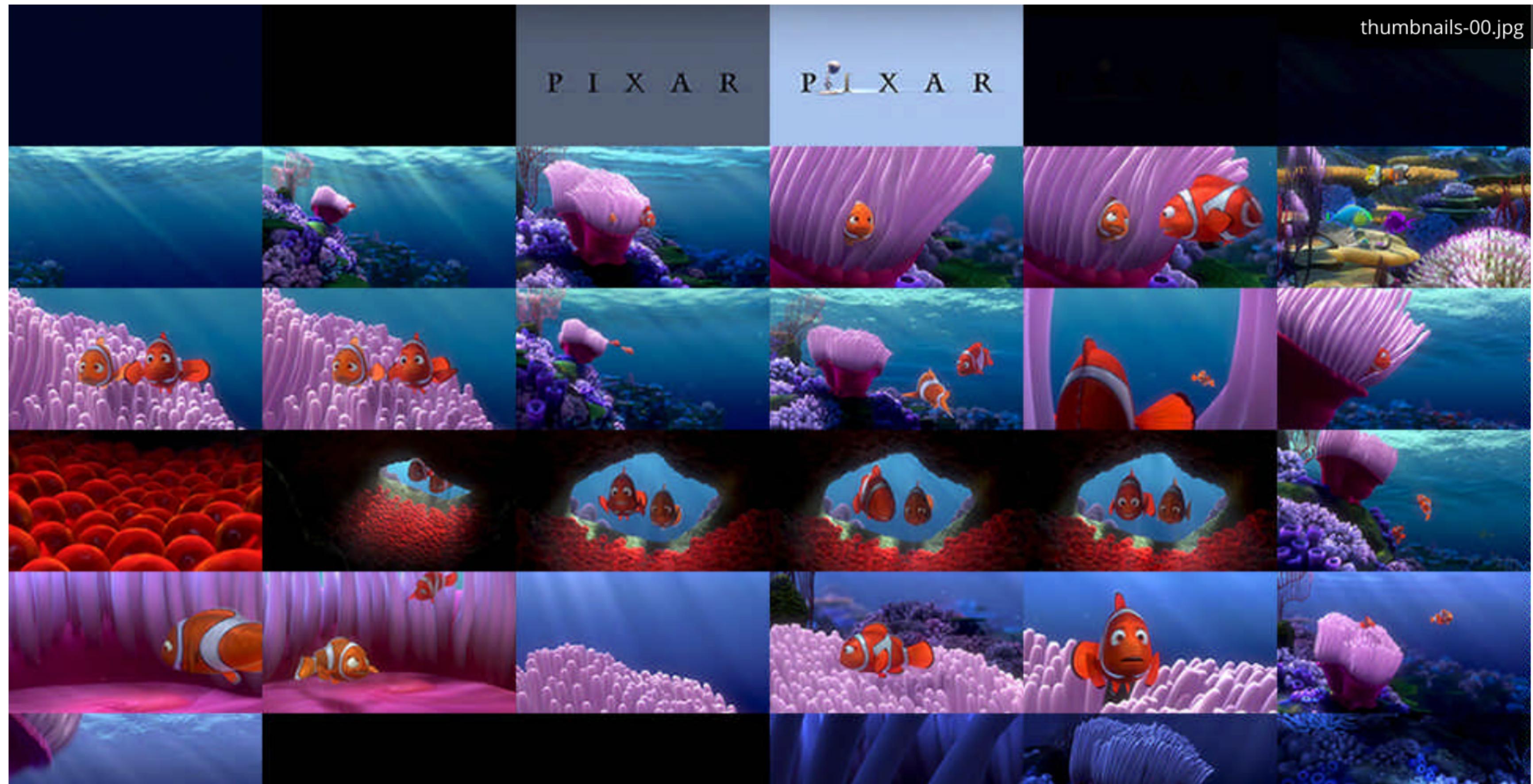
thumbnails-00.jpg?xywh=161,0,161,90

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:





Timelens

[Home](#) [Installation](#) [Usage](#) **Plugins** [About](#)

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 [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

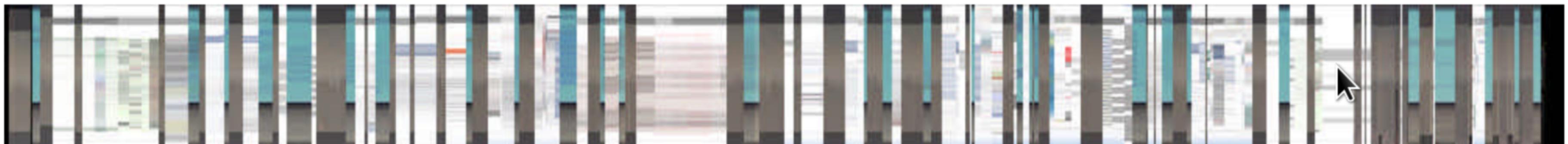
Suppose you have an empty div, which you want to make into a Timelens user interface:

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

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

```
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`:

```
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();  
  }  
});
```

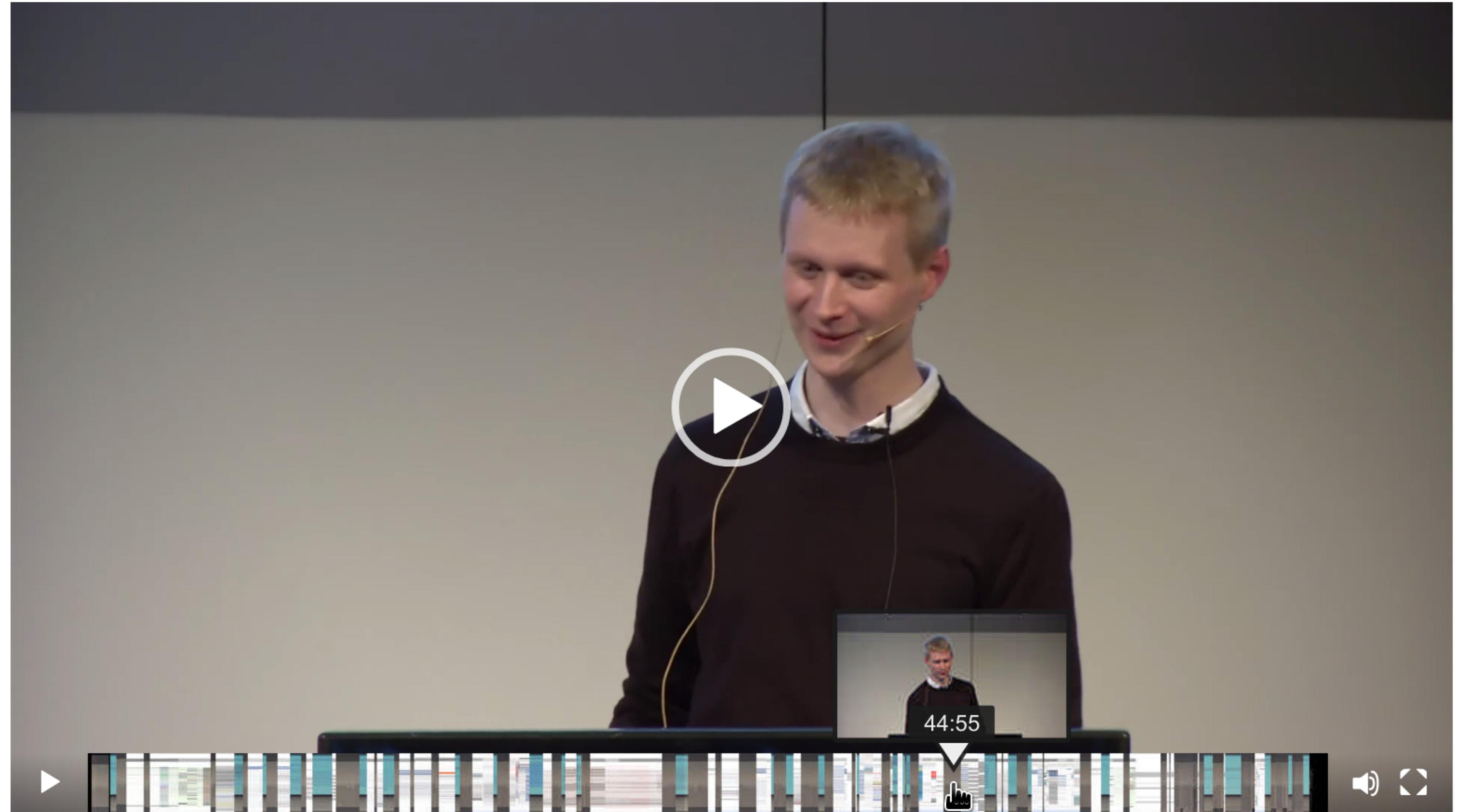
MediaElement.js

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"`:

```
<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:

```
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:

```
<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:

```
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"  
    }  
});
```

9. August: Nummernvertauschungen auf Defaults.

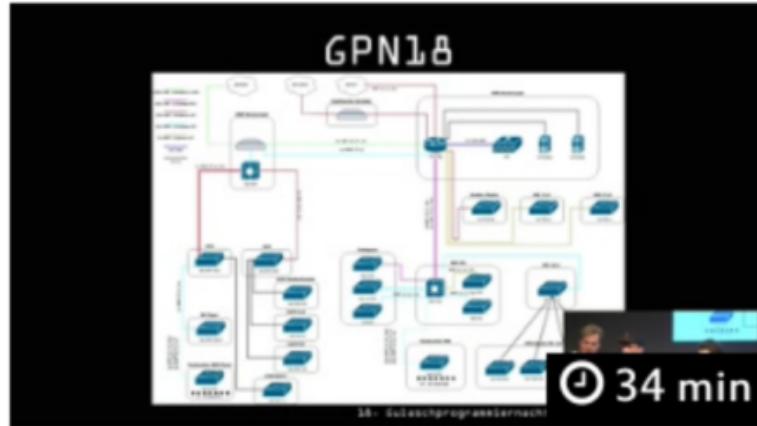




The Future



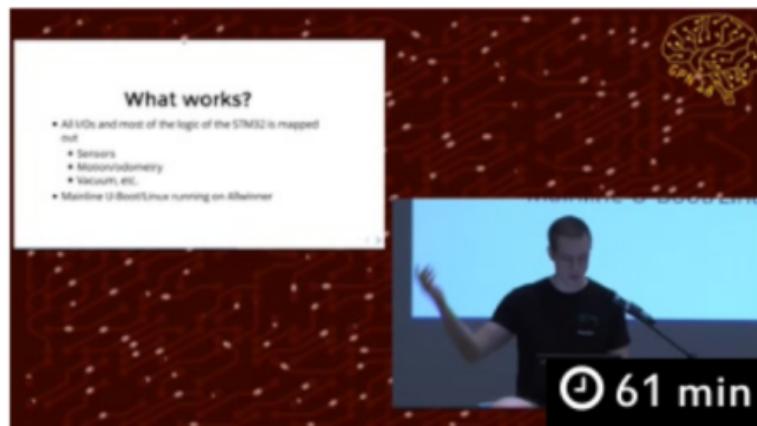
2018-05-12  42  Emma Lilliestam



37 min

Infrastructure Review and Closing

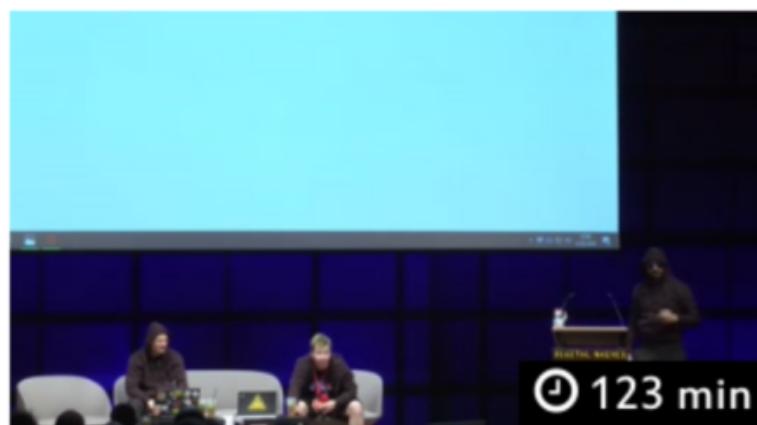
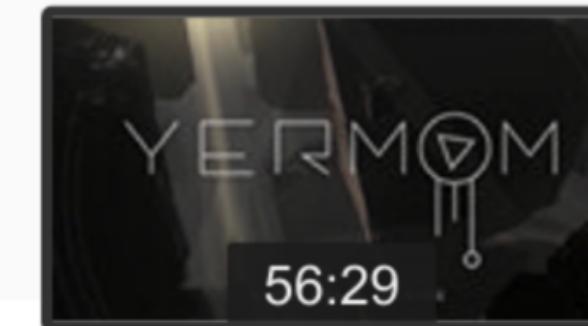
2018-05-13  279  obelix



61 min

Lightning Talks

2018-05-12  218  pony



123 min

Demoshow

2018-05-12  430  F1fth

https://api.timelens.io/0.1/youtube/XEVlyP4_11M





Secure | https://www.youtube.com/watch?v=XEVlyP4_11M

Print YouTube Vi

nordlicht.morr.co

timelens-youtub

...



Search



Edit bookmark

Name

URL

Bookmarks bar

Other bookmarks

Cancel

Save

Government Surveillance: Last Week Tonight with John Oliver (HBO)



gulasch



Timelens

Home **Installation** Usage Plugins About

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):

```
$ 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:

```
$ 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
```

Amarok - Hunting For Witches by Bloc Party

Engage Playlist Mode Tools Settings Help

Music Lyrics Artist Search: bloc

Context Collection Playlists Files Devices

Hunting For Witches - Bloc Party
A Weekend In The City

Track played 40 times
98 ★★★★★
Last played: 17 hours ago
First played: December 2006

Favourite Tracks by Bloc Party

Hunting For Witches	98	★★★★★
Song For Clay (Disappear Here)	98	★★★★★
Kreuzberg	97	★★★★★
Waiting For The 7.18	98	★★★★★
Sunday	97	★★★★★
I Still Remember	96	★★★★★
SRXT	97	★★★
The Prayer	97	★★★
Uniform	96	★★★
On	96	★★★

Albums by Bloc Party

A Weekend In The City	11 Tracks	2007	51:33
Silent Alarm	14 Tracks		1:04:42

Disc 1

01 Like Eating Glass (4:21)
02 Helicopter (3:40)
03 Positive Tension (3:54)
04 Banquet (3:22)
05 Blue Light (2:47)
06 She's Hearing Voices (3:29)
07 This Modern Love (4:25)
08 Pioneers (3:35)
09 Price of Gas (4:19)

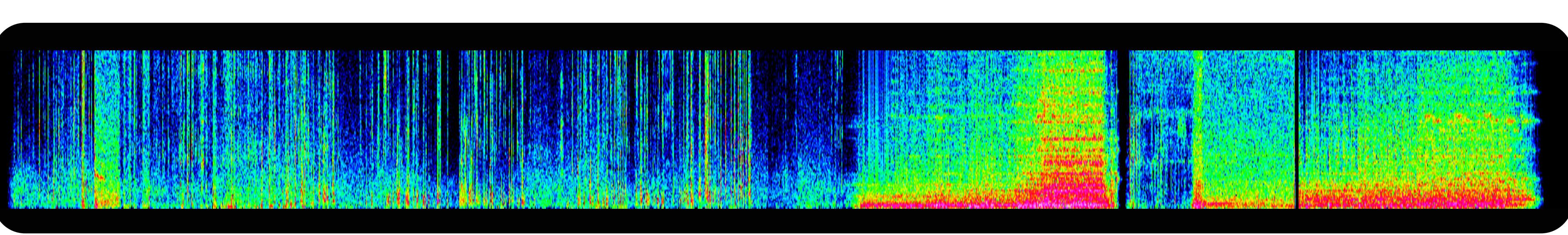
Playing: Hunting For Witches by Bloc Party on A Weekend In The City (3:31)

11 visible of 26 tracks (51:33) 0:19 -3:12

2 - Hunting For Witches (3:31)
Bloc Party - A Weekend In The City (98) (40)

100%

★★★★★



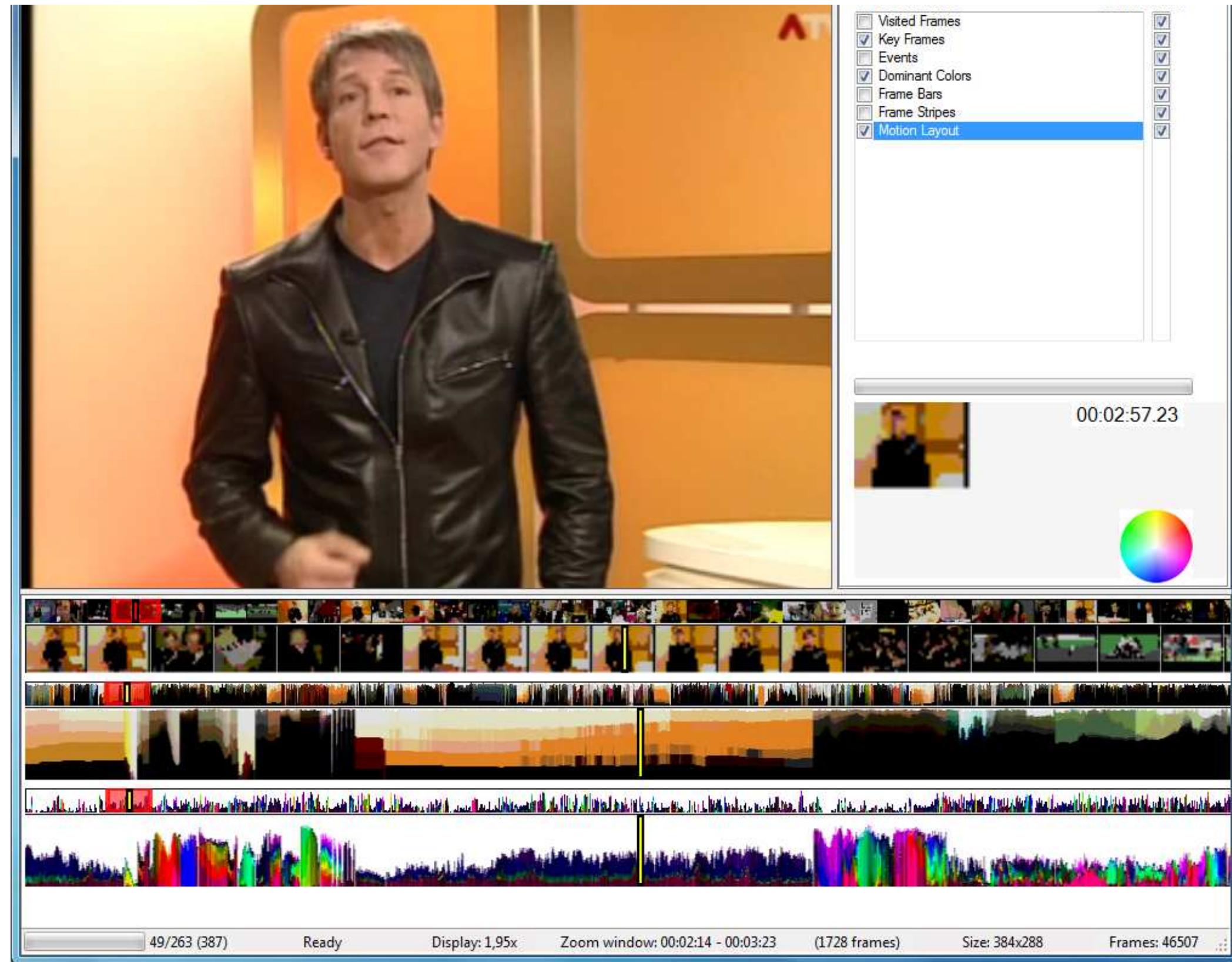


MacBook Pro



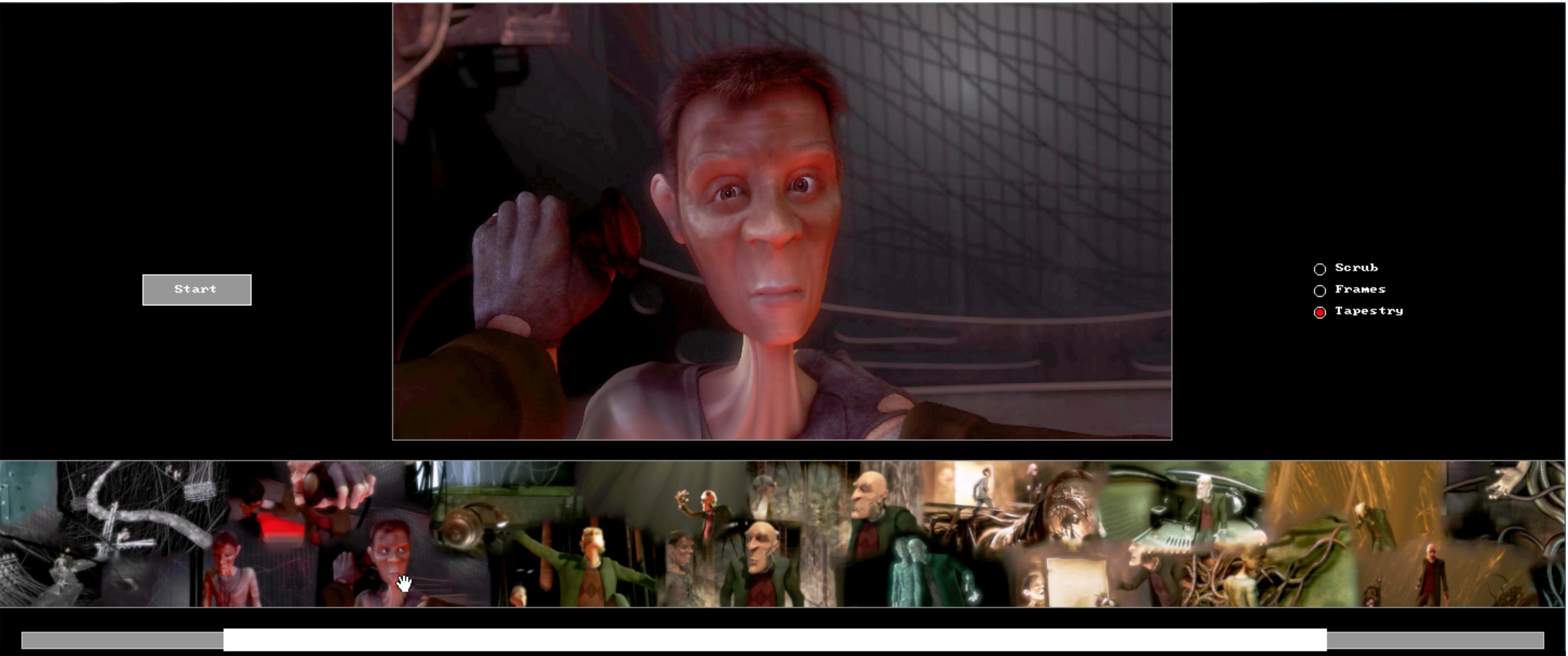
Interactive Navigation Summaries

Schoeffmann & Boeszoermenyi (2009)



Video Tapestries

Barnes/Goldman/Shechtman/Finkelstein (2010)



3D Filmstrip

Hudelist/Schoeffmann/Boeszoermenyi (2013)

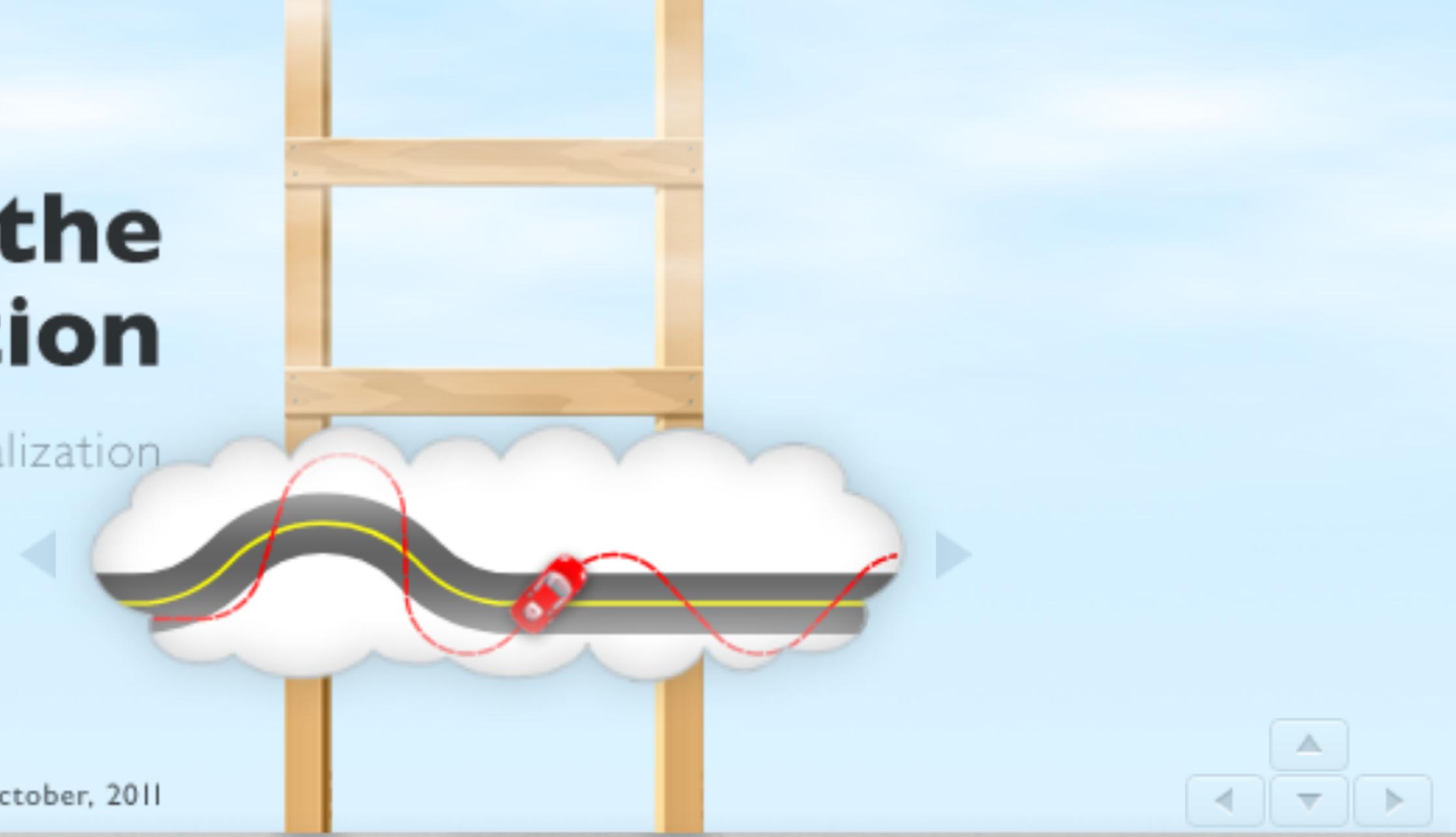


EXPLORABLE EXPLANATIONS

Up and Down the Ladder of Abstraction

A Systematic Approach to Interactive Visualization

Bret Victor / October, 2011



"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

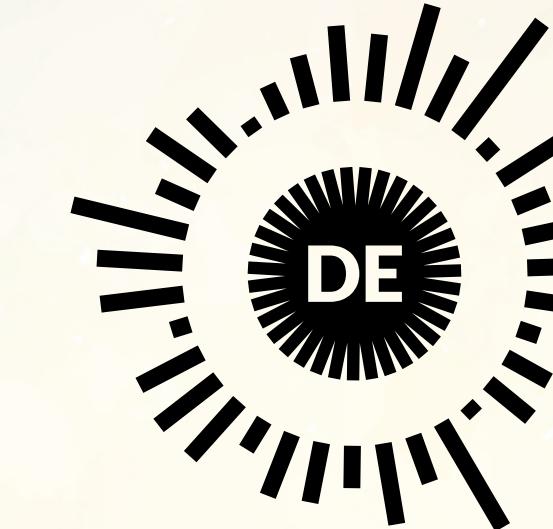


timelens.io
@timelens_io



Bundesministerium
für Bildung
und Forschung

P
Prototype
Fund



OPEN
KNOWLEDGE
FOUNDATION
DEUTSCHLAND