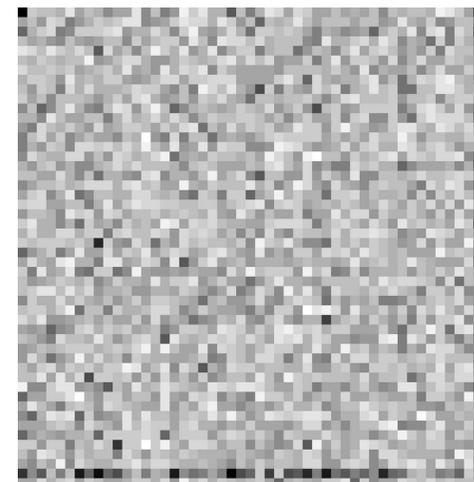


Reconstructing composites using source coordinates from ↗LOCK

Composites can be constructed in different ways, most commonly by just resizing each frame of a video into a single pixel and then creating an image by placing those left-to-right, top-to-bottom, with the width determined on a case-by-case basis. Here instead, rather than the resizing method, arbitrary coordinates have been selected as the pixel to composite the frames from.

The motivation for this experiment was Discord user *Ferran_RM11* generating a new keyhole heat-map for ↗LOCK, exposing a *X* structure that resembled a *X marks the spot* treasure map (shown to the right). With ↗LOCK (together with ↗DELOCK ↗RELOCK and ↗UPRIGHT) being thought of as critical components of the UFSC corpus, as well as the keyholes (the individual pixels that tend to show up at random in UFSC videos), it seemed reasonable to extract interesting coordinates from the heat-map and use them as maps of their own to work with other corresponding, classic 50x50 videos.



Three sets of coordinates were extracted, for a total of 11 coordinates:

Main	All the fully black pixels (no keyholes)	0:0 0:48 48:22
Extended	Permutations of coordinates from the main set.	0:22 22:0 22:22 22:48 48:0 48:48
More	Another near-zero value and the bottom-most, right-most pixel near the 48:48 “X”.	48:8 49:49

The 28 videos examined were, for a combined total of 308 new composites:

```
574016 BRILL_B CFO CRIMP DELOCK DIFF GEN HARVEST JALVINSACH MAX_TEND PORT RATE REVI STEM BRILL_49999 BRILL_N COEF  
DDR_DONE DIAGONAL FOLD GOLDEN HULL LOCK NEO RAPID RELOCK RUN UPRIGHT
```

Conclusion

While nothing particularly remarkable was discovered in this experiment, some composites stand out, which are listed next. If nothing directly related to the experiment itself, the mystery of keyholes continues to intrigue. as studying these composites serves as a reminder of their arbitrary nature – while also often being the only “moving part” compared to what could be construed as the “background” of [the more typical composites](#).

Continued study must be performed on keyholes: if a decoding method is ever found, their prevalence in videos of this particular format would result in a remarkable amount of recoverable data.

✂BRILL

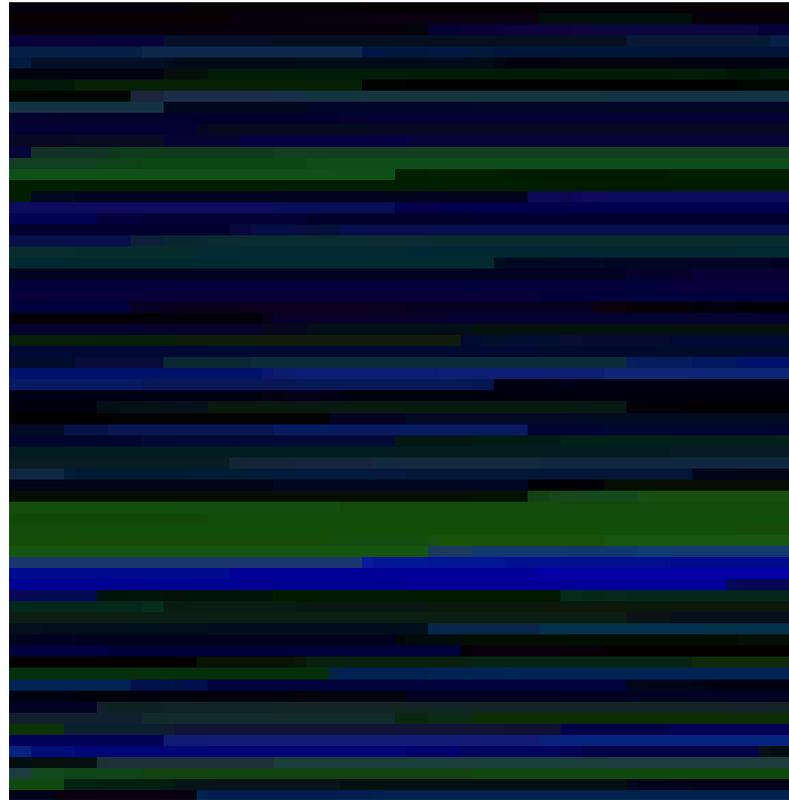


As composite 48:22 shows, at first glance ✂BRILL seems to be a static image, with some movement only at a certain point in the video...

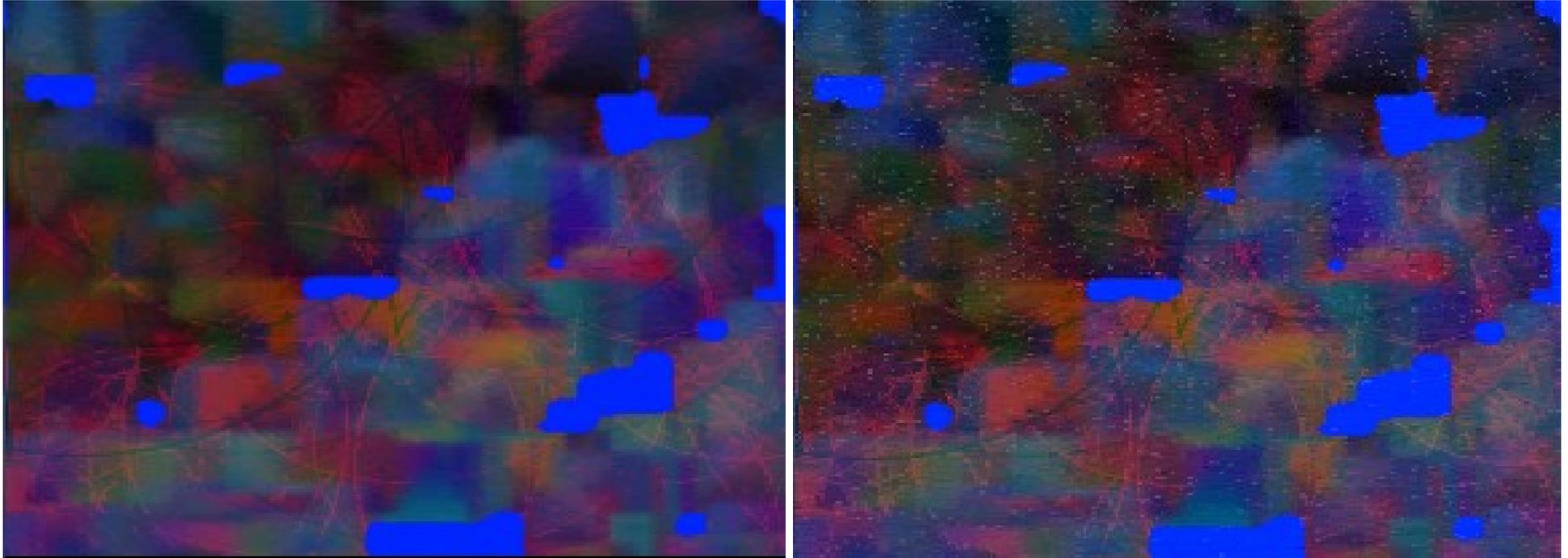


.. but what about this barely-noticeable pattern shown in coordinate 48:0? If this wasn't a static image, it could perhaps be explained by compression artifacts but, that not being the case. is this an intended feature by UFSC?

(Note: the pattern is so subtle some PDF readers won't show it. If the image doesn't have a faint striped pattern, try using another reader).



Given ↗DELOCK's nature, picking different coordinates to produce composites from will yield vastly different results – however, one thing rarely mentioned or studied about the video is that its movement isn't linear all of the time but rather has some jumps and freezes, much like some sort of streamed video trying to catch up after moments of latency. This is represented well by the 22:22 composite: its broad bright green-blue segment seems to indicate a moment when the original video freezes in its motions twice in a row before resuming as normal.



Composite from [the Visual Overview wiki page](#).

The origin coordinate $0:0$ exhibits an unusual amount of keyhole activity, much like with ⌘LOCK (see introduction). Also notable is the fact that many of these are wide, with 2-plus pixel widths – uncommon in the scope of UFSC in general, compared to most keyholes in other videos.

⊕RATE



YouTube thumbnail.



22:22 coordinate.



48:22 coordinate.

⊕RATE escapes the usual classic format of 50×50 videos in the fact that it looks more like an animated graph than random colors and pixels on the screen – however, as can be seen here, by picking certain coordinates to composite from, we can extract typical keyhole behavior from what initially looks like a completely different format.