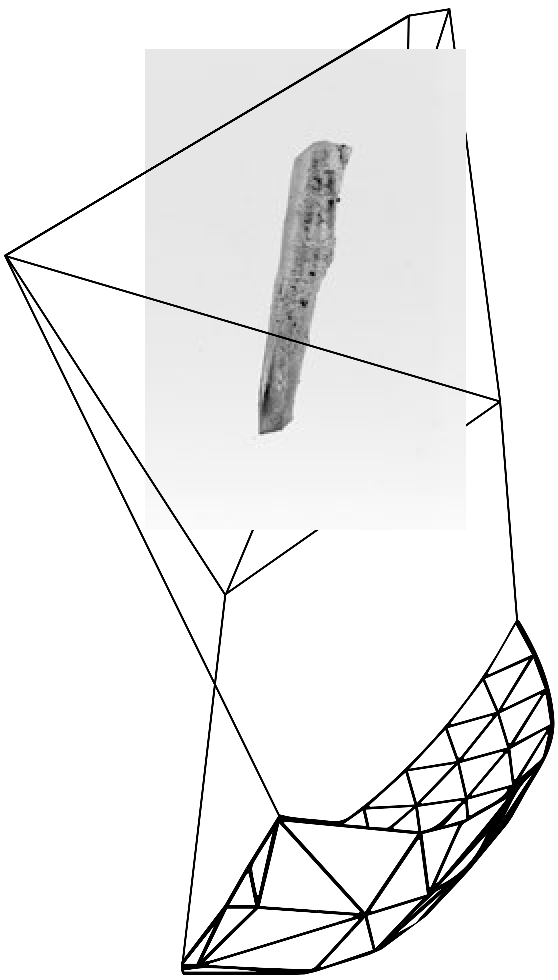
Form & Contagion

Formal Enquiries into Networked Media Interfaces



Jesse Josua Benjamin

"A commonsense is a topography of the common, determining what objects are given as common objects, what spaces are visible as spaces for discussion about common objects, what subjects are counted as able to perceive those objects and to make statements and decisions about them."

/

Jacques Rancière

This publication showcases one set of experimental prototyping based on recorded interaction with a networked media interface. This is part of a larger research project whose aim is to question what is currently considered to be a natural divide between temporary abstract data and visible information in 21st century media.

Form follows function: in the design of digital media this dictum has come to understand the latter as the purely visual function, in which the user navigates an interface as an aspect of the functionality itself. Whereas Peter Behrens and subsequent designer-engineers sought to configure a new commonsense of equality by stripping products of all functionally unadhering ornament, socio-economic conditions rarely manifest so visually in digital media. Instead, they are obfuscated beneath overly rationalized interfaces.

Thus, networked media interfaces are primarily ornaments to complex computational operations that distribute various contents. In the interface, these are rendered sensible if the conditions of the network are fulfilled: a wager on the user's interaction with the rendered content within a media space highly saturated with advertisements.

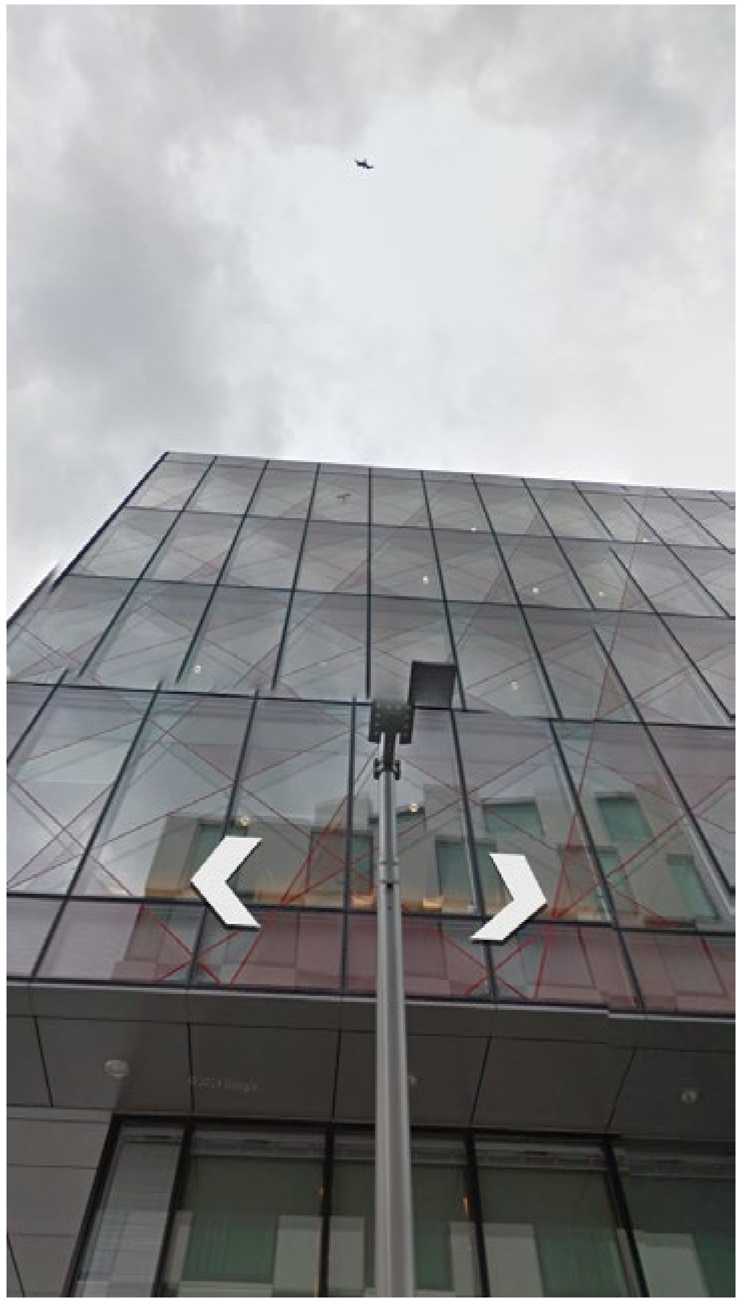
It is at the level of this distribution that a contagion of the commonsense is facilitated by (1) the socio-economic intentions of the network and (2) the ontological conditions of computational processes. To be precise, even if emotional contagion—i.e. the influence on users by exposure to specific types of content—occurs with intent by the company running a particular networked media, it is itself infected by abstract algorithmic objects.

That is, if "offline-behaviour" is influenced by orchestrated exposure to specific content on networked media interfaces, the orchestration itself is a careful management of infinite varieties of data and their straggling for attaining the status of sensible information.

Author's networked media account
as seen via Rafaël Rozendaal's
abstractbrowsing.net

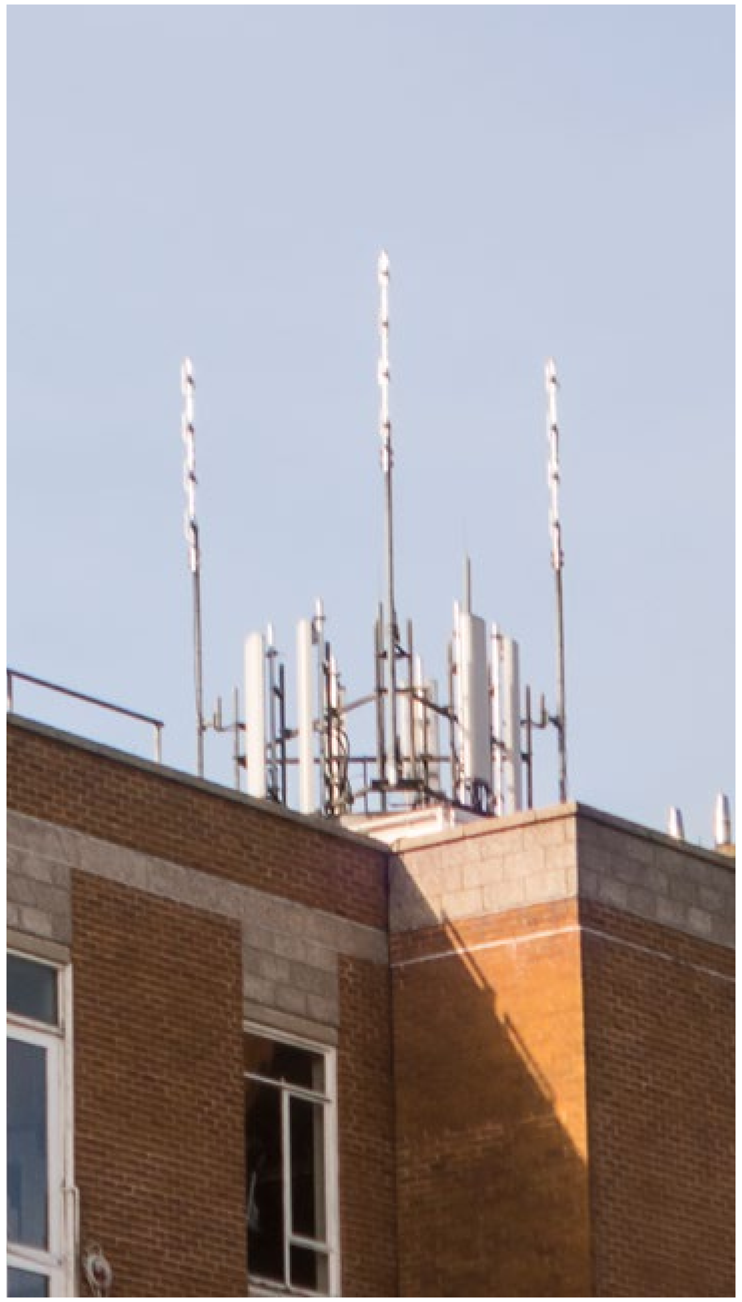
Furthermore, the classic material socio-economic conditions of networked media are, at least, two-fold. Both can be extrapolated from the IP-address of the active user. This governs, through the relation to other users, firstly the content that is directly associated with the active user; simultaneously linking it with a geographic location. This second link is of no less importance, as it influences what type of commercial, political and cultural content is displayed beyond the user's immediate consent or activity.

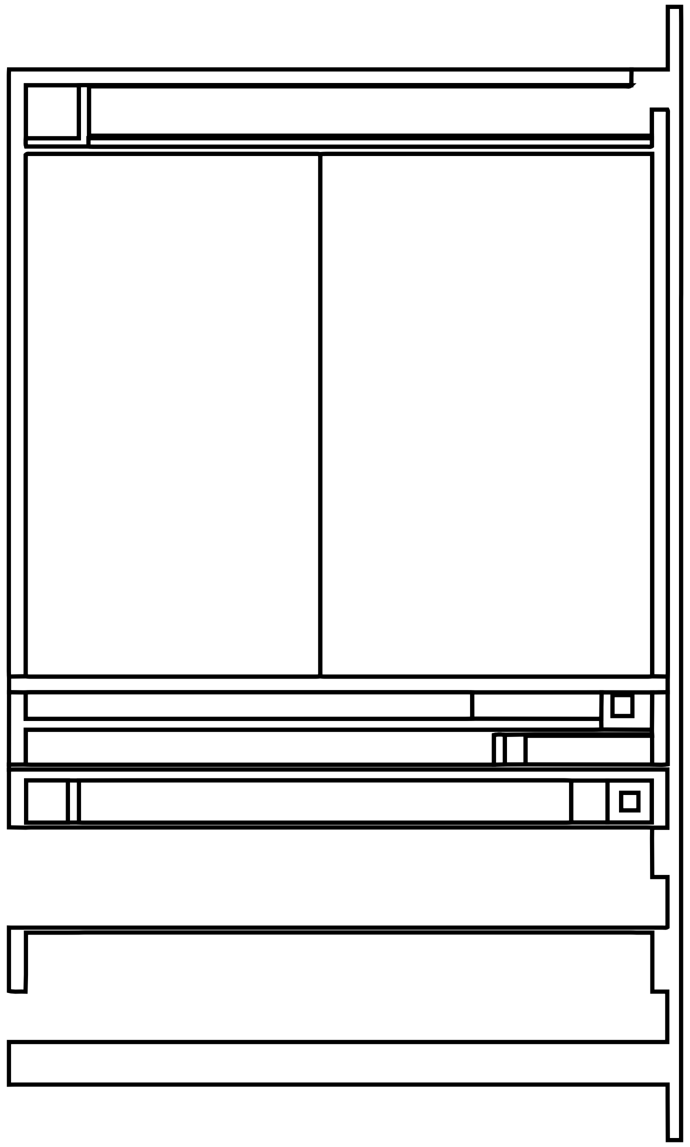
Remote IXP (1)



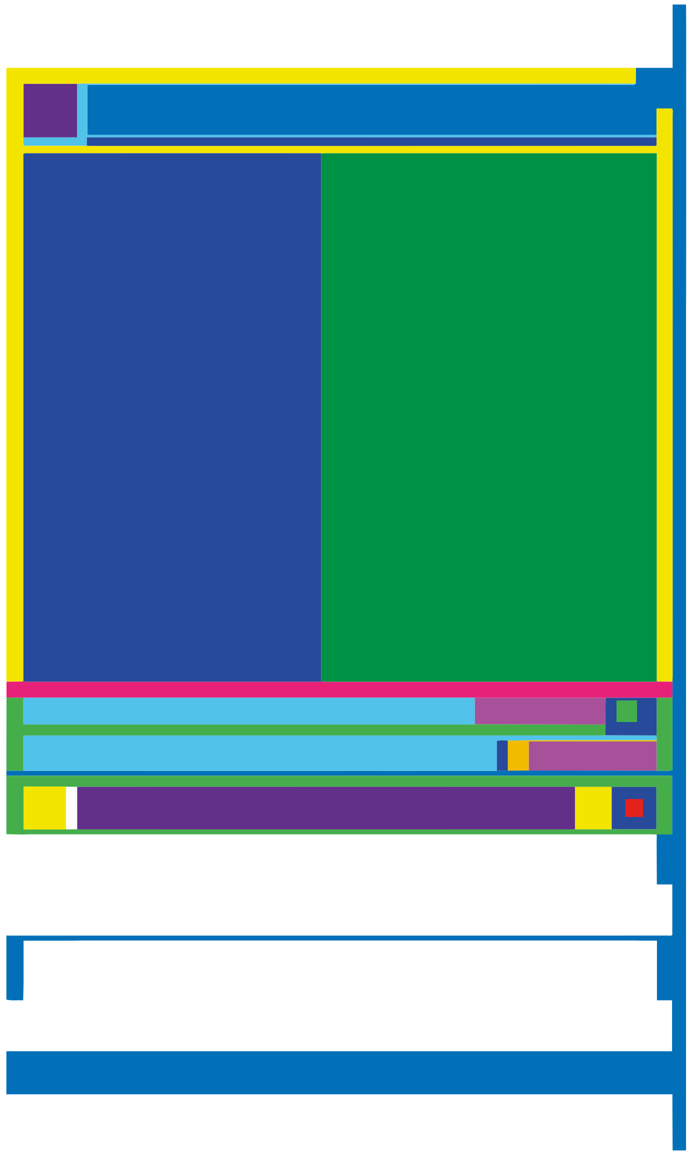
Secondly, the traditional material relation to (1) the remote content-supplying internet exchange point (IXP) of the network and (2) the local IXP. These two facets of the materiality, i.e. the contagion of the commonsense and the geographical relation necessary to facilitate it, form a new understanding of the functionality of networked media.

Local IXP (2)





Potential Content

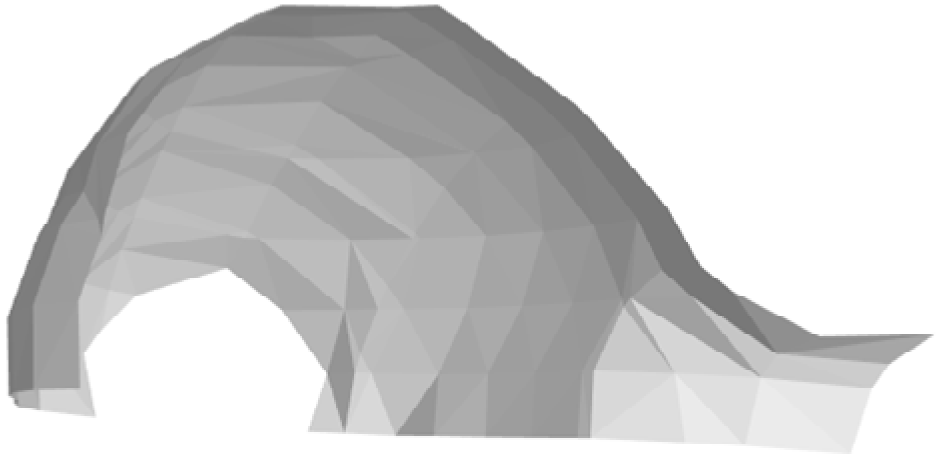


Actual Content

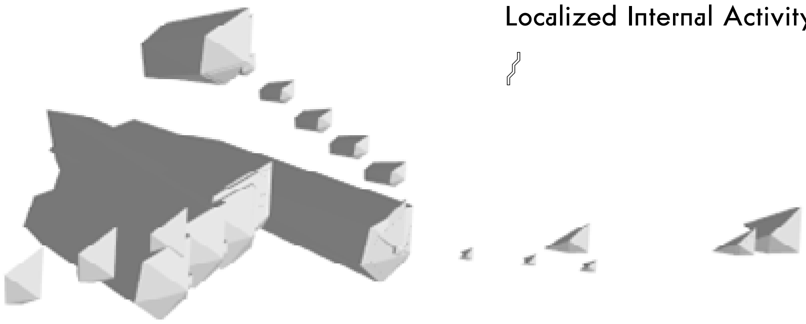
Translating this functionality into form balances precariously on the boundary between what is considered raw data on the one side and information on the other. In the recorded data the existence of this boundary becomes all the more evident, while at the same time potential for it's removal is equally available for discovery.

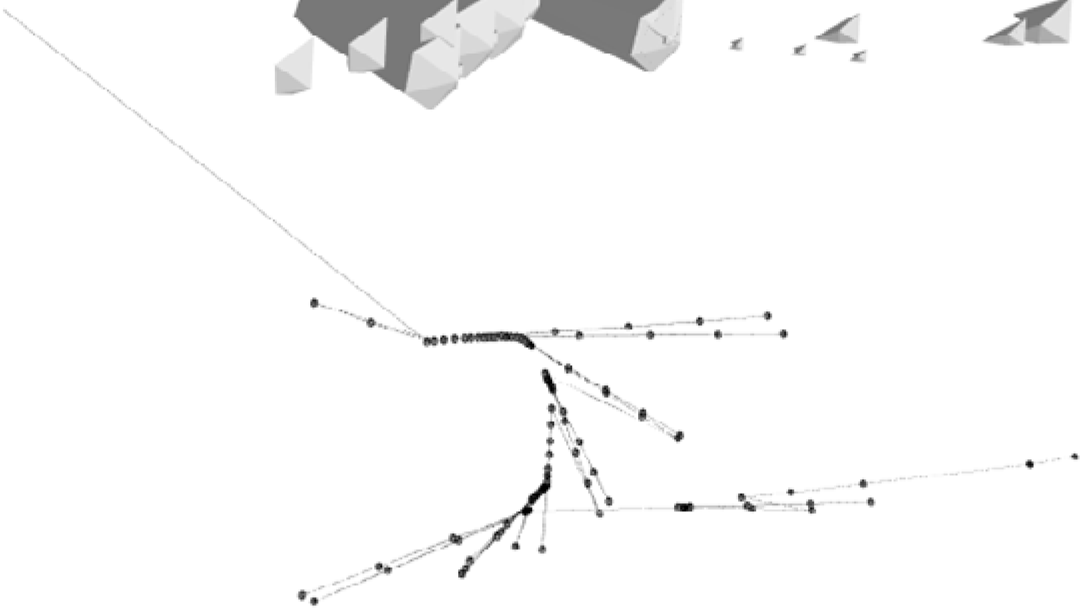
For example, the recorded data shows that the networked media operates using SyntheticEvents, meaning events held at the ready to produce and visualize content in case the network and user interaction necessitate it. Computational prototyping generates forms that can be associated with these aspects of networked media.

Topology for Parameters (3)

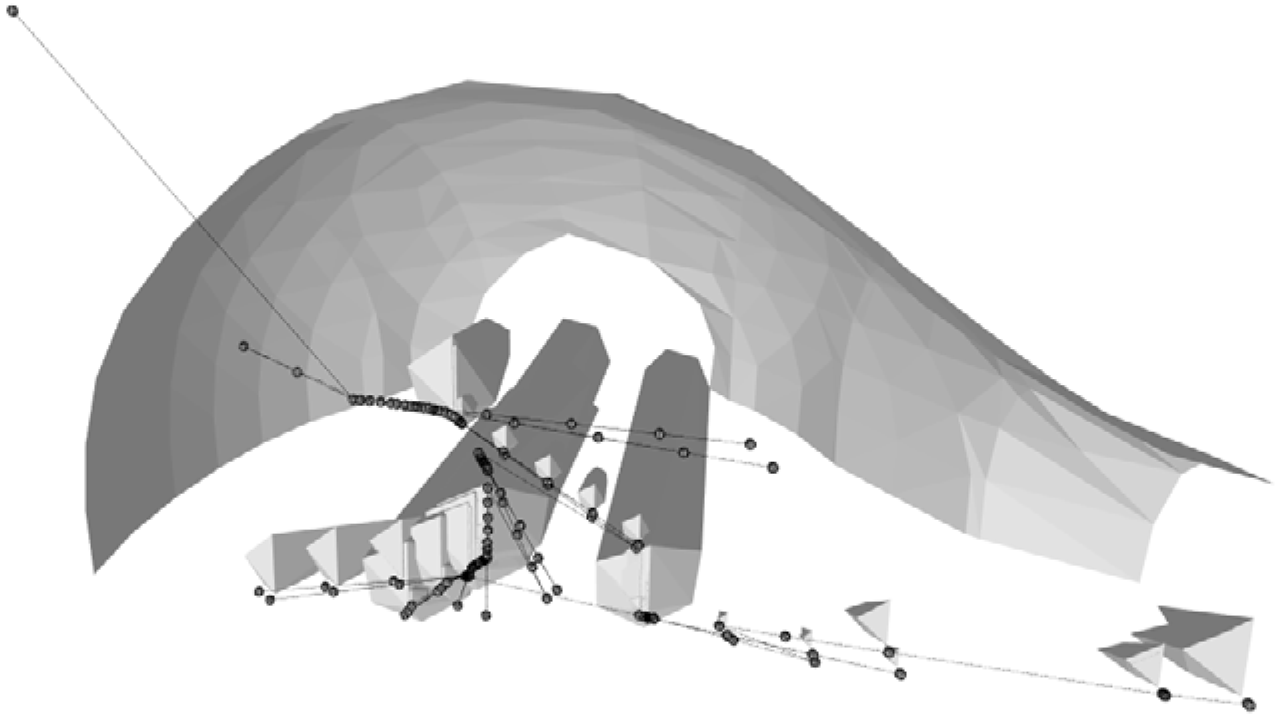


Localized Internal Activity (2)





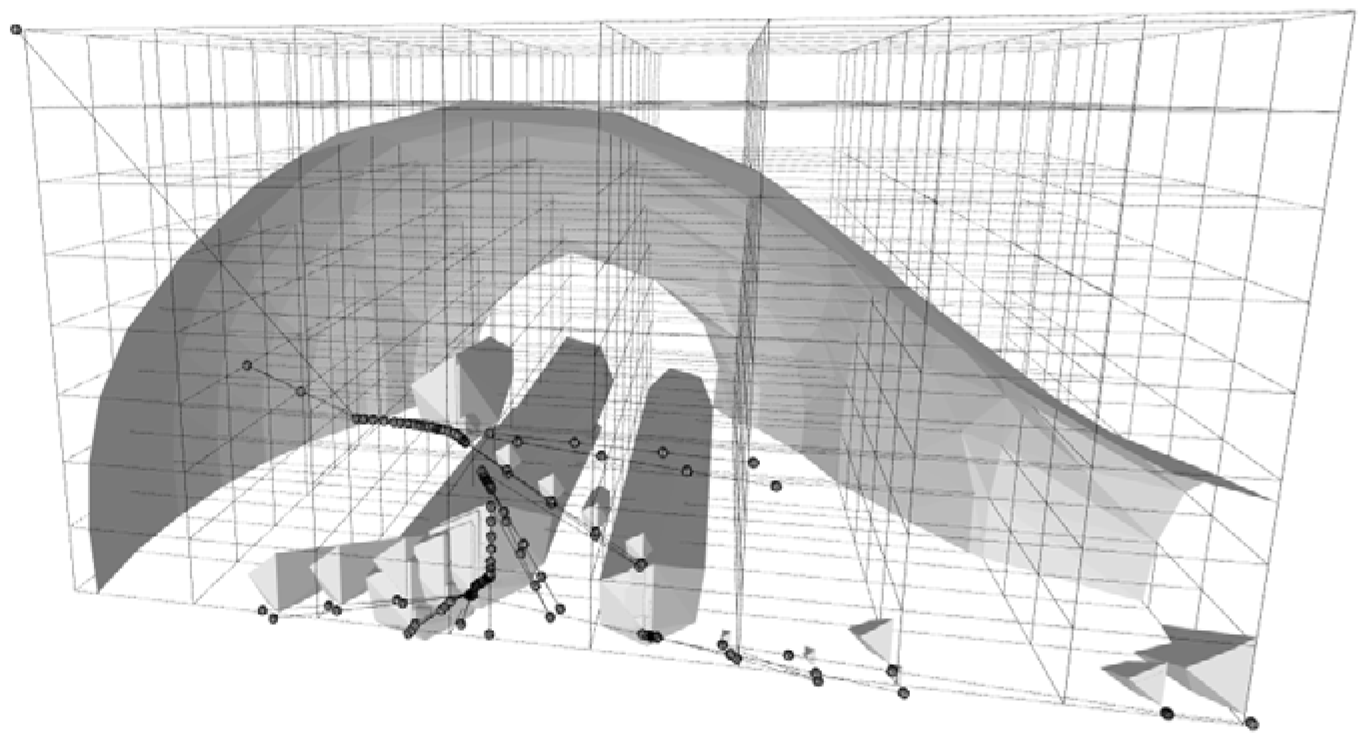
Interface External Activity (1)



What is sensed as external by networked media—in this case, cursor movement—is related to (1) the various instances of content on the interface, (2) the computational processes relating to each instance and (3) the overall spatiotemporal topology that relates to each computation.

Practically, this means creating a four dimensional representation of the networked media interface for a given time span: using the recorded coordinates of the cursor with the corresponding time allows endowing temporary abstract data with imageable properties.

```
public class Configuration {

  ArrayList<Comparative> comp;
  ArrayList<IsoWrap> isos;
  ArrayList<IsoSurface> surfaces;
  ArrayList<PVector> XYs;
  ArrayList<PVector> XYZs;

  int surfaceFill = 360;
  int surfaceStroke = 10;

  color bw = color(0);

  boolean wireframe = true;
  boolean fill = false;

  private PApplet pa;

  Configuration(PApplet _pa) {
    pa = _pa;
    comp = new ArrayList<Comparative>();
    isos = new ArrayList<IsoWrap>();
    surfaces = new ArrayList<IsoSurface>();
    XYs = new ArrayList<PVector>();
    XYZs = new ArrayList<PVector>();
  }

  public void initSpatiotemporality(PVector p, int depth, int asynch, int timeDOM)
{
    XYs.add(p);
    if (depth > 4) {
      FloatList xPts = new FloatList();
      FloatList yPts = new FloatList();
      FloatList zPts = new FloatList();
      for (int j = 0; j < depth; j++) {
        PVector mod = new PVector(
        p.x + (map(j, 0, depth, -(asynch*.001f), (asynch*.001f)))*(sin(j)*PI),
        p.y + (map(j, 0, depth, -(asynch*.001f), (asynch*.001f)))*(cos(j)*PI),
        p.z - ((j)*(asynch*.001f))
          );
        xPts.append(mod.x);
        yPts.append(mod.y);
        zPts.append(mod.z);
        comp.add(new Comparative(mod, timeDOM));
        XYZs.add(mod);
      }
      xPts.sort();
      yPts.sort();
      zPts.sort();
      createSurface(xPts, yPts, zPts, timeDOM);
    } else {
      return;
    }
  }

  public void createSurface(FloatList _xPts, FloatList _yPts, FloatList _zPts,
int _timeDOM) {
    IsoSurface _iso = new IsoSurface(
    pa,
    new PVector(_xPts.get(0), _yPts.get(0), _zPts.get(0)),
    new PVector(_xPts.get(_xPts.size()-1), _yPts.get(_yPts.size()-1), _zPts.
get(_zPts.size()-1)),
    8);

    for (Comparative c : comp) {
      for (int i = 0; i < _xPts.size (); i++) {
        PVector spaceTemp = new PVector(_xPts.get(i), _yPts.get(i), _zPts.get(i));
        if (c.isOverlap(spaceTemp) && c.isCongruent(_timeDOM)) {
          _iso.addPoint(spaceTemp);
        }
      }
    }

    surfaces.add(_iso);
  }

  public void constructIsos() {
    PVector[] vertices;
    int getter = 0;

    for (int i = 0; i < surfaces.size (); i++) {
      vertices = surfaces.get(i).vertices;
      getter = floor((vertices.length)/2);
      for (int j = 0; j < vertices.length; j++) {
        for (int k = 0; k < comp.size (); k++) {
          PVector x = vertices[i];
          if (comp.get(k).isOverlap(x)) {
            surfaces.get(i).addPoint(new PVector(
                        comp.get(k).p.x,
                        comp.get(k).p.y,
                        comp.get(k).p.z));
          }
        }
      }
    }
  }

  public void displayTopology(float _scale, float _isoThreshold) {
    scale(_scale);
    if (!wireframe) {
      fill(surfaceFill);
      stroke(surfaceStroke);
      strokeWeight(.5f);
    } else {
      noFill();
      stroke(surfaceStroke);
      strokeWeight(.5f);
    }
    for (IsoSurface s : surfaces) {
      if(showVoxels) {
        s.plotVoxels();
      }
      s.plot(_isoThreshold);
    }
  }

  public void displayColumns(float _scale) {
    lights();
    scale(_scale);
    if (!wireframe) {
      fill(30, 70);
      noStroke();
    } else {
      noFill();
      stroke(0);
      strokeWeight(.5f);
    }
    for (IsoWrap i : isos) {
      i.plot();
    }
  }

  public boolean hasRelation(Comparative _c, PVector _p, int _t) {
    if (_c.isOverlap(_p) && _c.isCongruent(_t)) {
      return true;
    } else {
      return false;
    }
  }
}

//--------------------------------------------

class Comparative {
  PVector p;
  int t, d;
  int asynch;
  float dist;

  Comparative(PVector _p, int time) {
    p = _p;
    t = time;
  }

  public boolean isOverlap(PVector _cP) {
    dist = p.dist(_cP);
    if (dist < 30) {
      return true;
    } else {
      return false;
    }
  }

  public boolean isCongruent(int cTime) {
    asynch = abs(t - cTime);
    if (asynch < 20000) {
      return true;
    } else {
      return false;
    }
  }
}
```

/ Consequently, by correlating the time-based identification of all recorded computational processes with the spatial data of all recorded cursor positions:

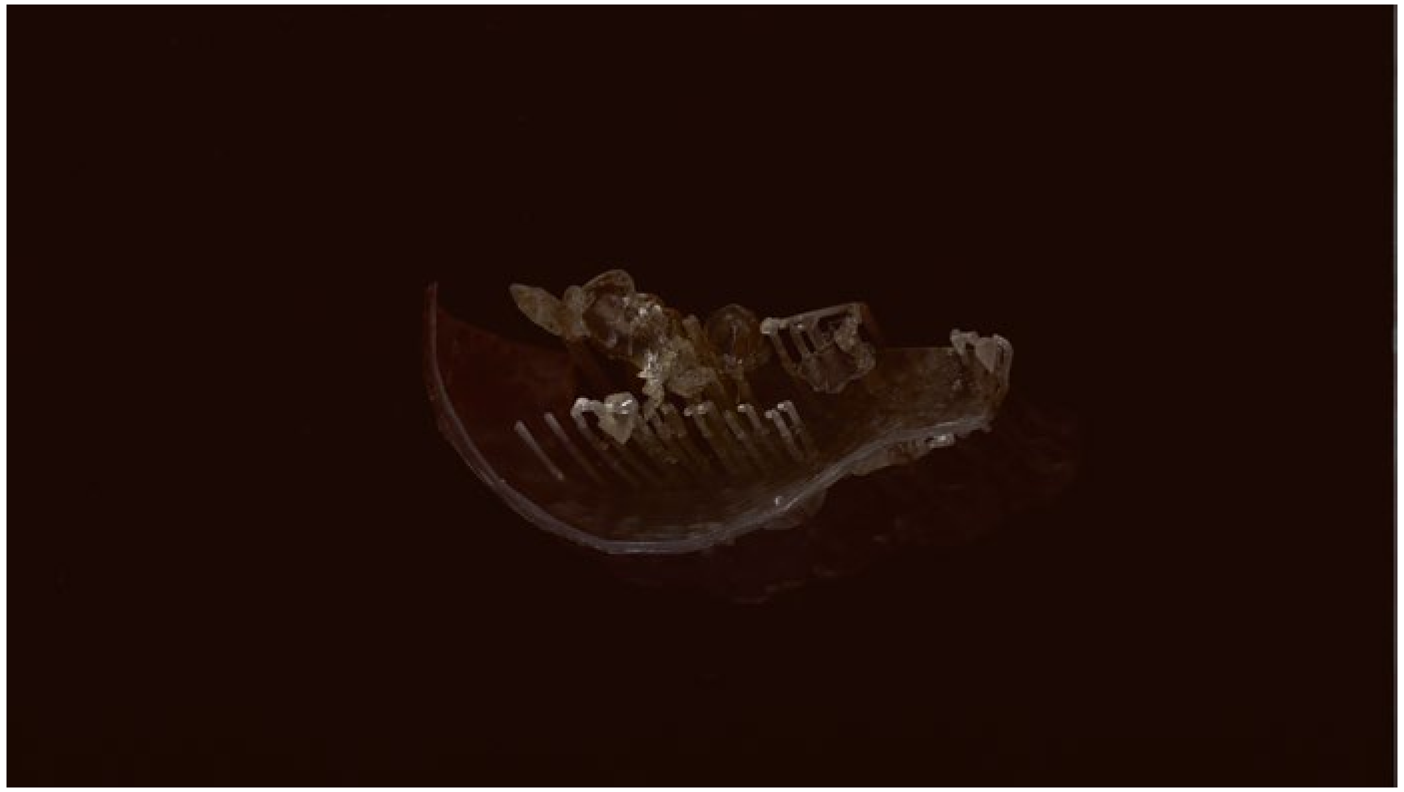
spatioTemporalPoint = cursorPosition & minimumTimeDistance

an overarching topology can be described. Therefore, the translations inherent to networked media—e.g. between data and information—lend themselves to form experimentation and investigation.

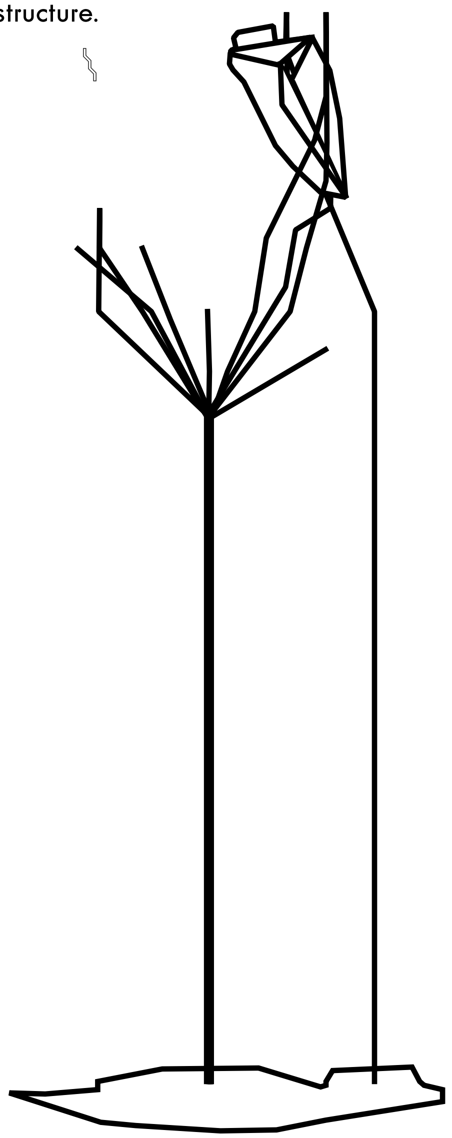
In a further step of formal experimentation, rapid prototyping is used to provoke forms alien to the original networked media interface.

In this process, the automatically generated support structures illustrate the imperceivable gaps between the functional parameters of networked media. Stereolithic printing software judges a computed model on terms of spatial integrity and rotates it accordingly. As such, it is as impartial as possible when it comes to the formal additions to the model. In their translucent visceral aesthetic, these prints render the complexities of the computational processes of networked media imageable and relatable.

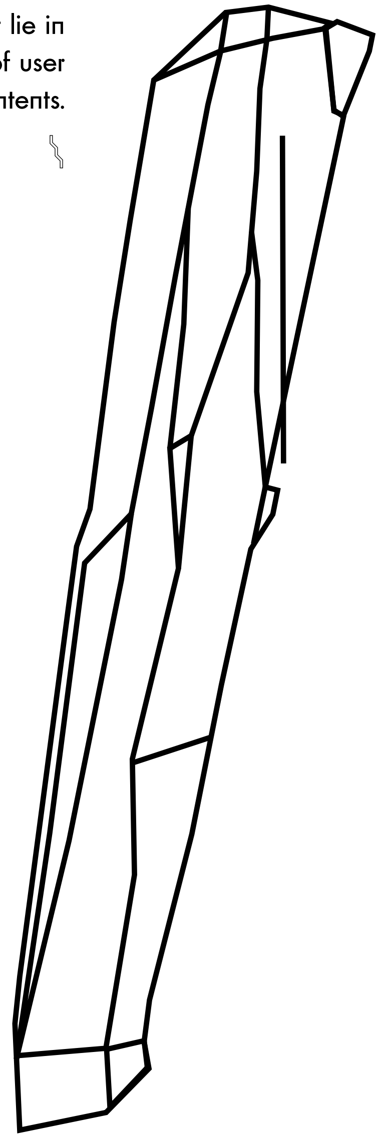
Photopolymer Resin Print

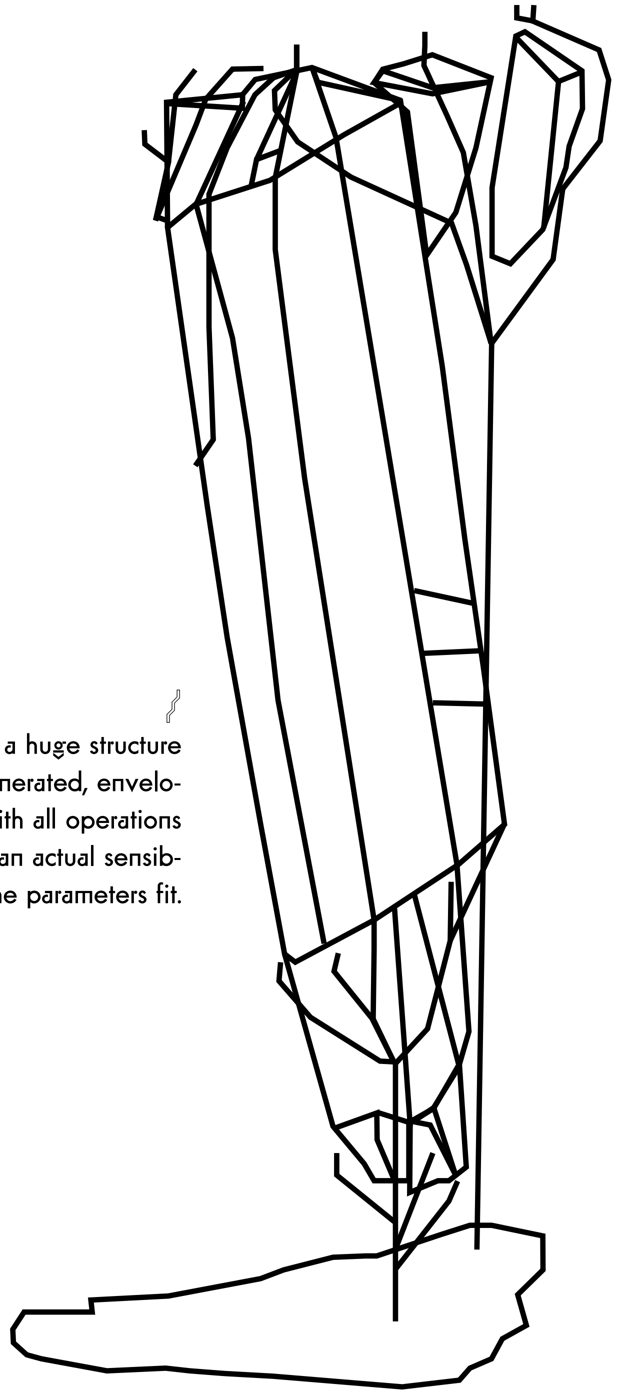


The automatically generated support structures illustrate the imperceivable gaps between the topology of parameters: the random data infecting the control infrastructure.

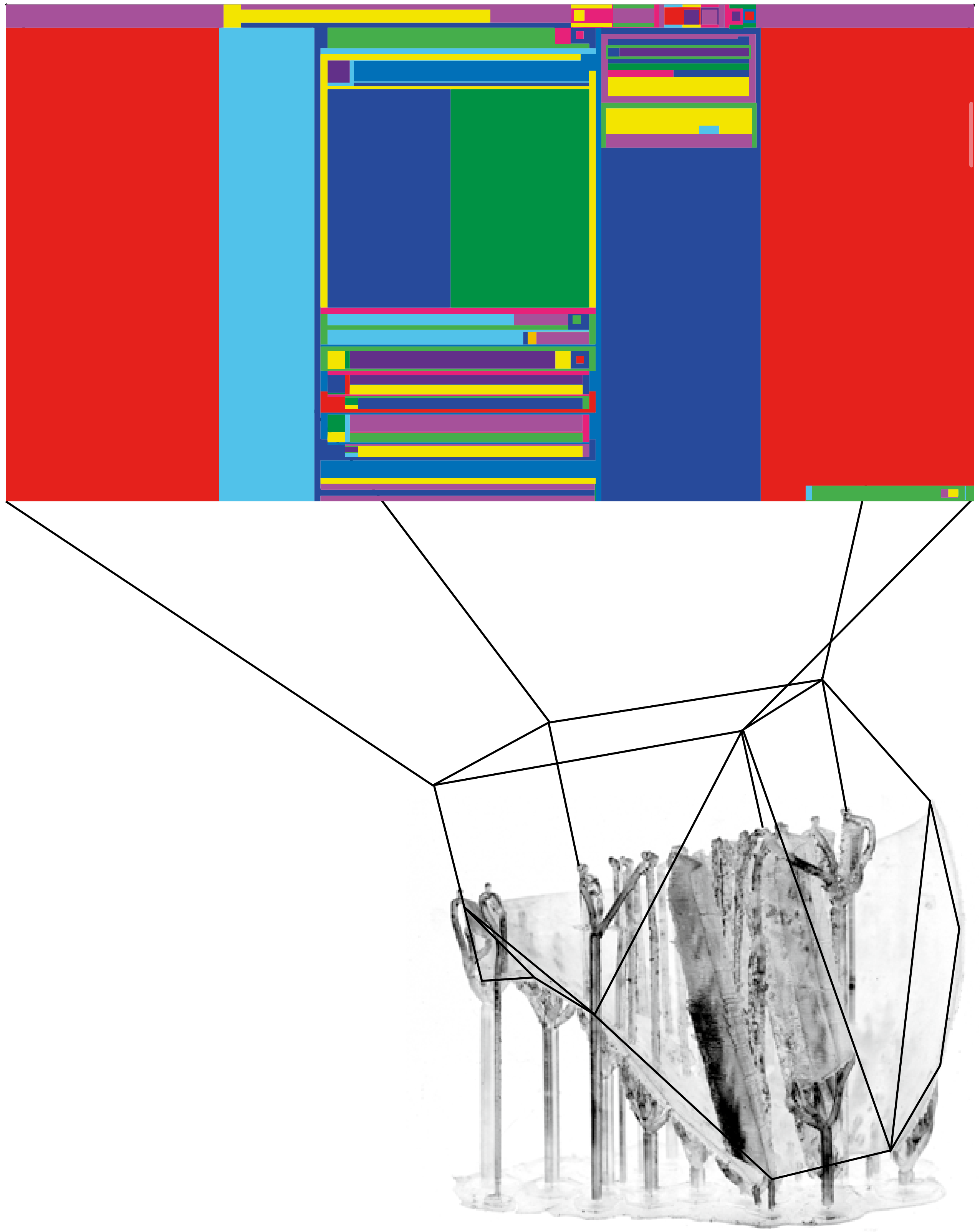


The SyntheticEvents held ready by the network's algorithmic infrastructure are actual but pre-sensible objects that lie in waiting for the right circumstances of user interaction and network contents.

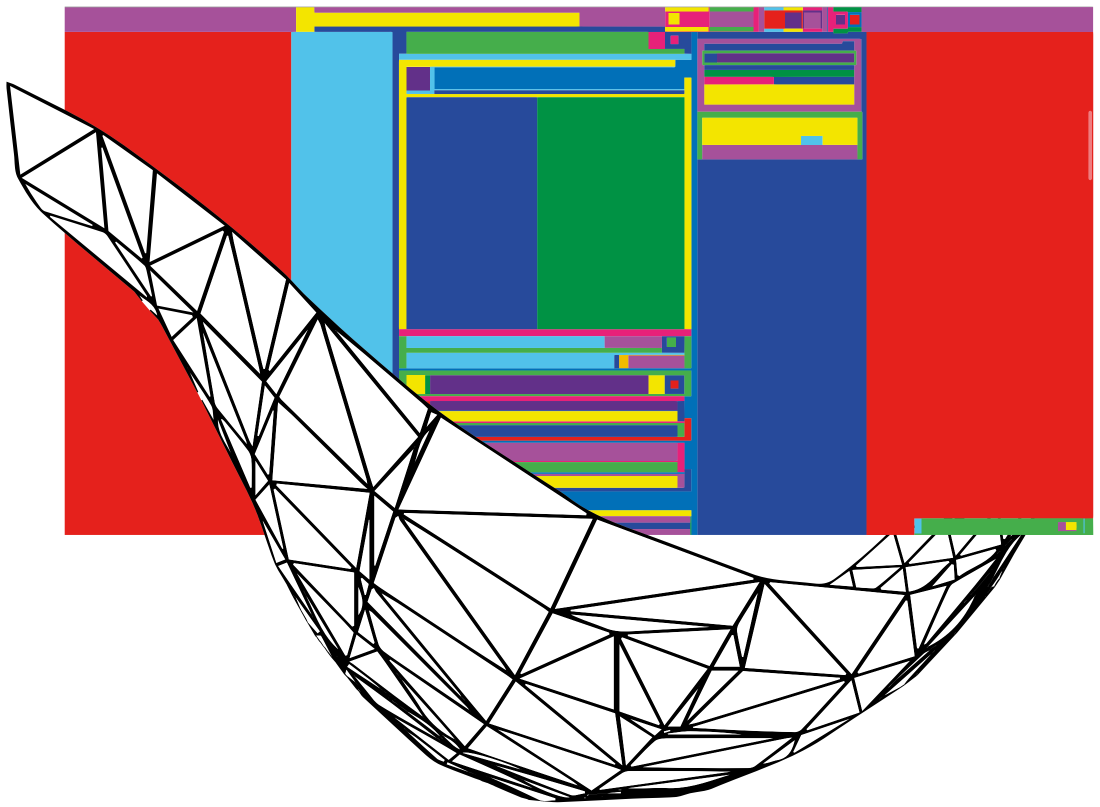




At the time of the click a huge structure of potential content is generated, enveloping a SyntheticEvent with all operations necessary to deploy it as an actual sensible object if the parameters fit.

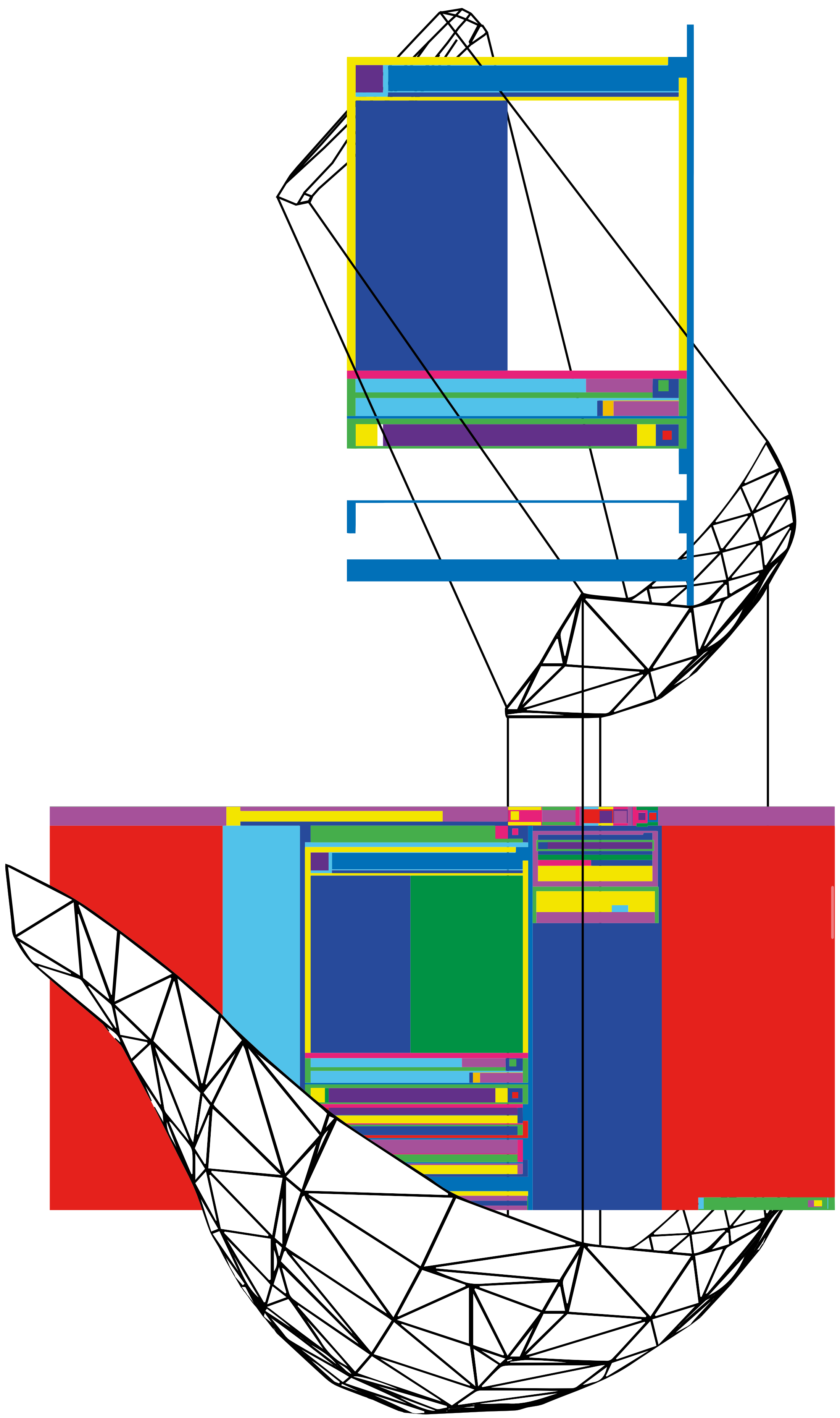


The collection of forms shown thus far are, while possibly intriguing, inherently abstract. If these were, as a thought experiment, to replace or become integrated in the networked media interface the case could surely be made that these forms obfuscate even more than they supposedly reveal. However, a spatiotemporal control structure exists which can be rendered visible independently of the boundary of the original interface.



/ The political and ethical dimension of this set of forms relates to what modes of subjectivity can operate using networked media, and what the conditions for their operations are. Consequently, revealing the actual complexity of the networked media interfaces and its processes shows the potentialities left behind in the obfuscation of temporary and abstract data. These forms, however abstract, give the function of networked media its due.

Their function, the abstract readiness for alternative distributions of visual content, is what can be formalized as the site of investigation for a contemporary designer-engineer. Therefore, endowing this functionality with form provokes a rethinking of the supposed rationalities of user-centered design. By engaging the complexities not realized, designing configuratively can be described as a way to reconfigure the commonsense constructed in 21st century media.



"It is representation in form, but not in deed, and this is the paradox. It is representation as mathematical recoding, not as any socially or culturally significant process of figuration, yet at the end of the day what emerges is exactly that."

/

Alexander Galloway

Galloway, A. R. 2012. The interface effect, Cambridge, UK ; Malden, MA, Polity.

Hansen, M. B. N. 2015. Feed-forward: on the future of twenty-first-century media, Chicago ; London, University of Chicago Press

Parisi, L. 2013. Contagious architecture : computation, aesthetics, and space, Cambridge, Massachusetts, The MIT Press.

Rancière, J. 2007. What Does it Mean to be Un? Continuum, 21, 559-569.

Edward XIV
Special Edition
21/04/2016