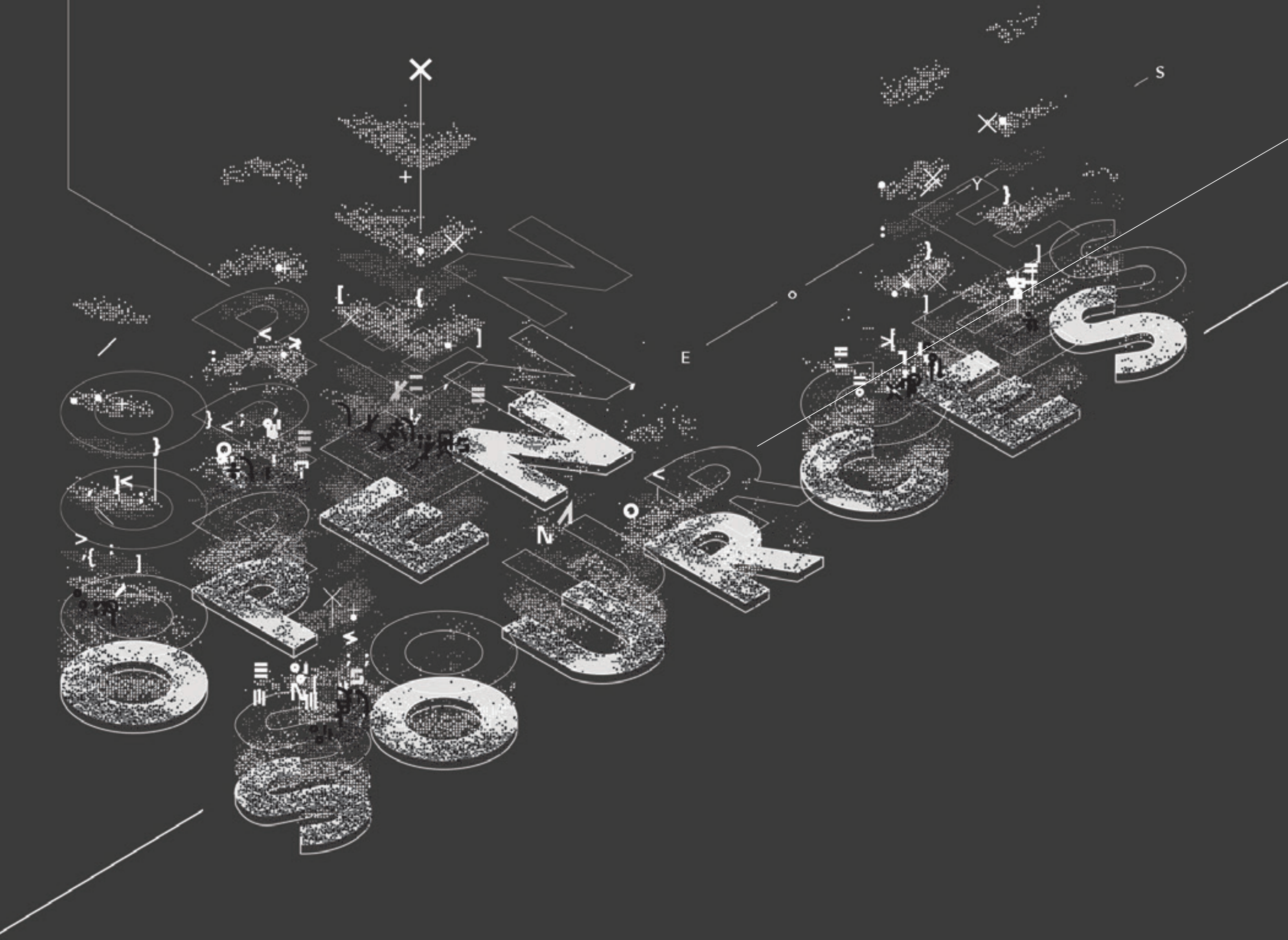
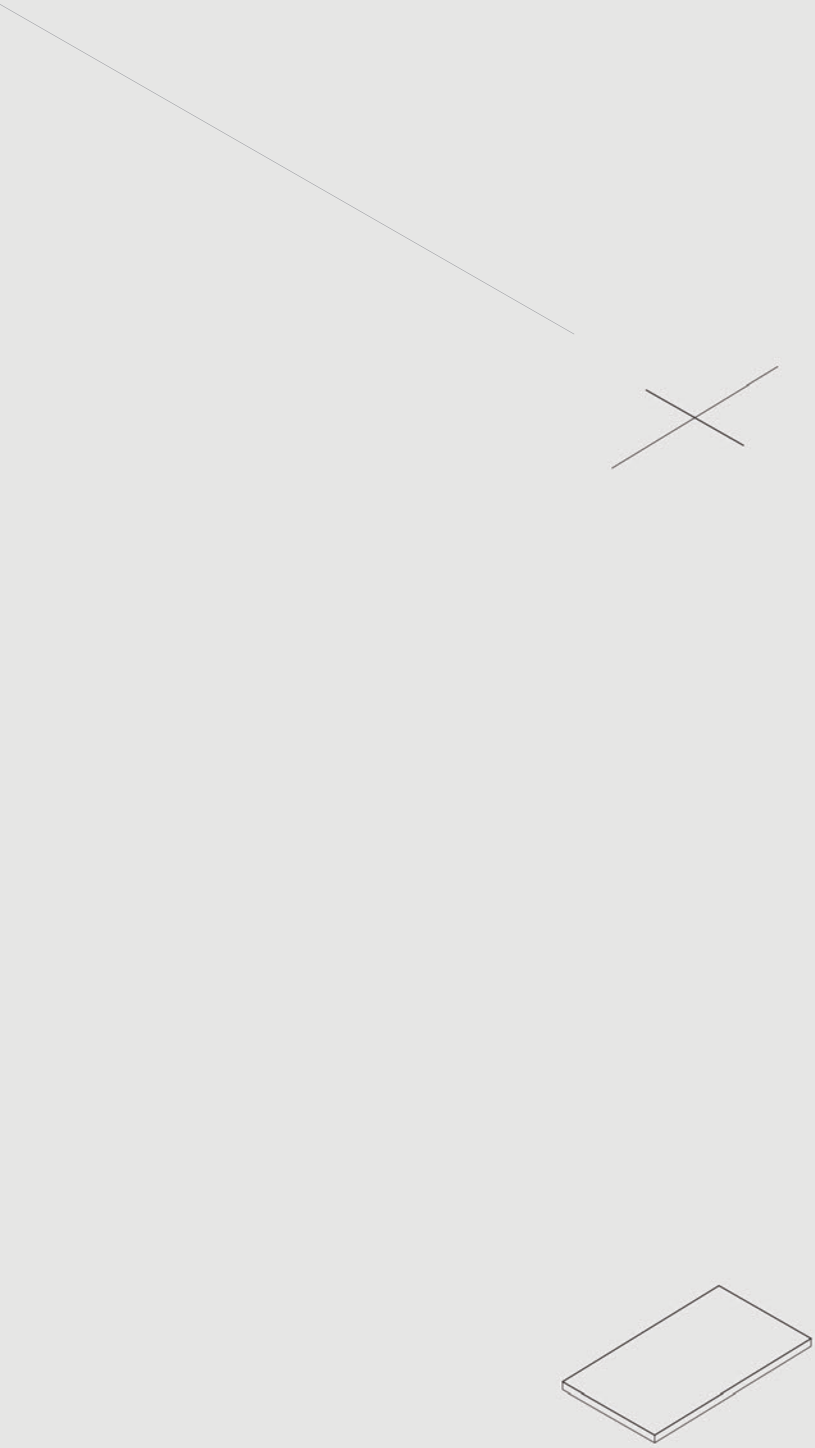
MAT Media Arts and Technology Graduate Program



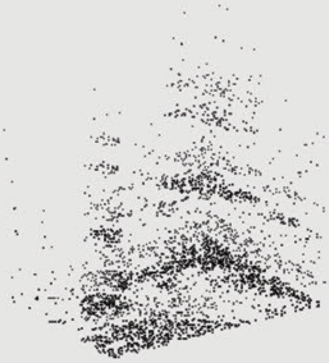
Juan Manuel **Escalante** • Fabian **Offert**

EDITORS



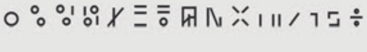
UNIVERSITY · OF · CALIFORNIA · SANTA BARBARA ·

LET THERE BE LIGHT

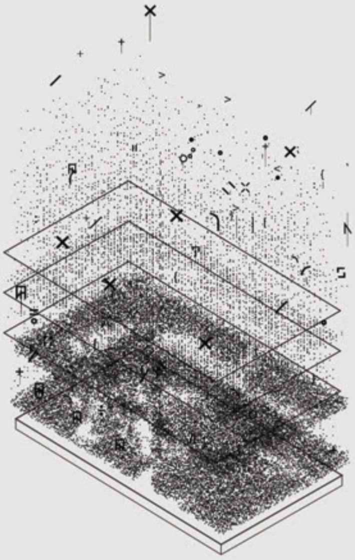


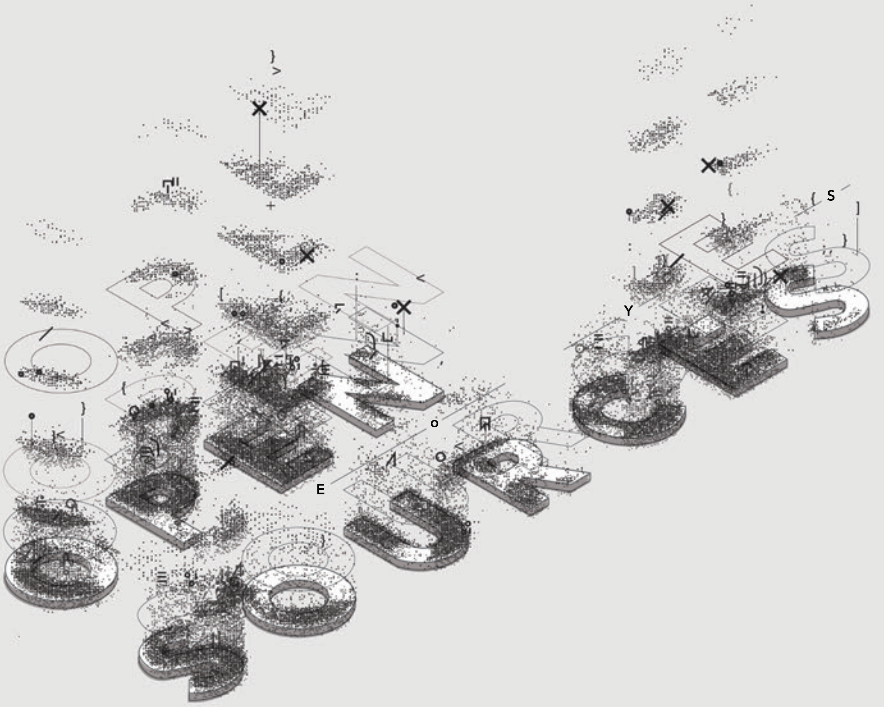
Media Arts and Technology
Graduate Program

{ } ; ' . + - // [ ]



\ × + ○ ●





Juan Manuel **Escalante** • Fabian **Offert**
EDITORS

DESIGN:
Juan Manuel Escalante

PRINT SUPERVISION:
Kurt Kaminski

Media Arts and Technology Graduate Program
University of California, Santa Barbara

All photographs and materials provided by their respective authors.
This is a non-profit publication.

MMXV

Contents

MAT FACULTY

ANDRES **CABRERA**
TOBIAS **HÖLLERER**
THEODORE **KIM**
JOANN **KUCHERA-MORIN**
GEORGE **LEGRADY**
MARCOS **NOVAK**
MARKO **PELJHAN**
CURTIS **ROADS**
MATTHEW **TURK**
MATTHEW **WRIGHT**

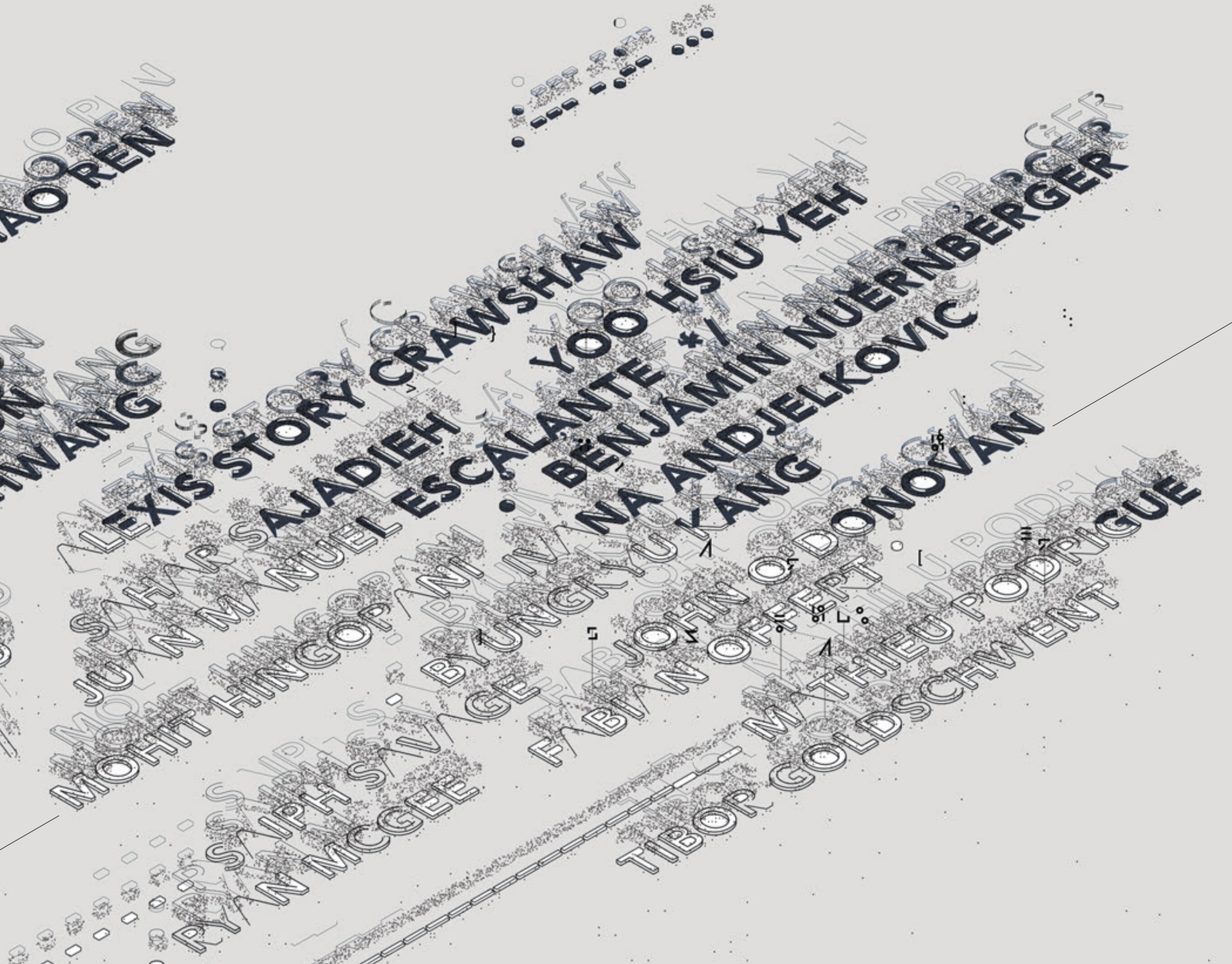
STAFF

LAURA **CHEUNG**
KRIS **LISTOE**
MARISA **ORTEGA**
LISA **THWING**
LARRY **ZINS**

EoYS TEAM

AKSHAY **CADAMBI**
JUAN MANUEL **ESCALANTE**
KURT **KAMINSKI**
JIELIANG **LUO**
INTAE **HWANG**
FABIAN **OFFERT**
MATTHIAS **WAGNER**
KEEHONG **YOUN**

( ) { KEEHONG YOUN
CECILIA WU
MUHAMMAD HAFIZ WAN ROSLI
CHRIS SWEENEY
MYLES SCIOTTO
KURT KAMINSKI
HANNAH WOLFE
CREATE ENSEMBLE
KARL YERKES
DONGHA
PABLO COLAPINTO
TIM WOOD
DANNY BAZO
DAVID GORDON
XIAOLI ZHAO
YILUN QIN
KUO-CHIN LIEN
INTAE H
NATALY MORENO


AO REN
WANG
LEXIS STORY CRAWSHAW
SAJADIEH
YOO HSIU YEH
MANUEL ESCALANTE
BENJAMIN NUERNBERGER
NA ANDJELKOVIC
JUSHAN
MOHIT HINGORANI
BYUNGKYU KANG
JOHN O'DONOVAN
FABIAN OFFERT
MATHIEU RODRIGUE
TIBOR GOLDSCHWENT
RYAN MCGEE

FOREWORD

/ + | ○ It is with great pleasure that I have the opportunity to introduce to you the first catalog of cutting-edge art and research produced in the Media Arts and Technology Graduate Program, realized by students for their annual End of Year Show, this year titled OPEN SOURCES. Both the exhibition and this book represent the best of what MAT has to offer and a glimpse into where our program is heading.

MAT is a unique program at the University of California, Santa Barbara, positioned at the intersection of arts, engineering, and science. Since its inception in 1999, MAT students and faculty have pushed the envelope of multi, inter, and transdisciplinary work in digital media.

MAT students come from a variety of backgrounds — most, though not all, have undergraduate degrees in a wide range of disciplines such as computer science, media art, architecture, electrical and computer engineering, music, and audio engineering. Many have hybrid backgrounds, with experiences and interests in both technical and aesthetic aspects of scholarship and creative work. The critical element, however, is a passion for thinking and working out of the box, beyond the traditional disciplinary boundaries, to create new media, new ideas, and new disciplines.

The creation of this book is an important step towards chronicling our activities. I would like to acknowledge the editors Fabian Offert and Juan Manuel Escalante, and the entire End of Year Show team, for pursuing and realizing this passion.

George Legrady

CHAIR · Media Arts and Technology Graduate Program

The Poetic Algorithm
or What Happened to Software Art?

FABIAN **OFFERT**

"Der poetische Algorithmus, von dem in den experimentellen Schreibweisen Gebrauch gemacht wird, kann nicht in jedem Falle herzbezwingend sein."[1]

Max Bense in San Francisco, 1969 | Credits: Elisabeth Walther-Bense, CC BY-SA 3.0 de

[1] **"The poetic algorithm, used by the experimental modes of writing, can not always be a matter of the heart."** Translated by the author.

These are Max Bense's[2] words in his book *Theory of Texts*,[3] a central building block within his lifelong project of reconciling logic and poetry, technology and aesthetics. Inspired by the rediscovered Gertrude Stein, as early as the late 1950s,[4] Bense and others dreamt the structuralist dream of an artificial poetry that had the immediacy of images, a poetry of pure (algorithmic) form. His prosaic verdict that such a poetry could not always be "a matter of the heart" mirrors its predicament. As early and contemporary critics never ceased to point out: Bense's artificial poetry has no windows to the outside world, its elements refer to each other, but to nothing beyond the text.[5]

The reception and reflection of scientific discovery in the arts often runs slow, with scientific paradigm shifts arriving in the art studio when the new paradigms have been overturned themselves already. It is highly unlikely that Bense, who studied mathematics and wrote a dissertation on the philosophical implications of quantum physics in 1938, would not have been familiar with Cantor's, Gödel's, and Turing's ideas on the incompleteness of formal languages,[6] so his persistence on the beauty of the exact, the structure, and the absolute, might be, in part, attributable to precisely this historic disparity between the disciplines, in combination with the then highly exotic appeal of using computers to produce, or even only help in the production of, aesthetic artifacts. Much more, however, it points to the important fact that Bense's project was not so much concerned with the results, but with the modalities of automated aesthetic production, with the creation and evolution of computer programs for the creation and evolution of computer-generated text, with poetic algorithms, and not with algorithms for the creation of poetry.

[2] Max Bense (1910-1990) was one of the most prominent protagonists of information aesthetics. A philosopher, mathematician, and engineer, he is regarded as the founder of the so called Stuttgart school of German (generative) computer art in the 1960s. See also: Walther, Elisabeth. "Max Bense und die Kybernetik". http://www.netzliteratur.net/bensekybernetik.htm.

[3] Bense, Max. *Theorie der Texte.* Cologne: Kiepenheuer & Witsch, 1962.

[4] The first theoretical reflection on such a practice was most likely Theo Lutz' article "Stochastische Texte" in the magazine augenblick in 1959. See: http://www.netzliteratur.net/lutz_schule.htm.

[5] See, for instance the critique in Florian Cramers's 2006 dissertation *Exe.cut[up]able statements. Poetische Kalküle und Phantasmen des selbstausführenden Texts.* The German magazine DER SPIEGEL printed a scathing review of Bense's poetry as early as 1962: "Max Bense. Entwurf einer Rheinlandschaft". In: DER SPIEGEL 20/1962. http://www.spiegel.de/spiegel/print/d-45140157.html

[6] Cantor, Georg. "Ueber eine elementare Frage der Mannigfaltigkeitslehre". In: *Jahresbericht der Deutschen Mathematiker-Vereinigung*. Leipzig, Stuttgart, Wiesbaden, 1892. Gödel, Kurt. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: Monatshefte für Mathematik und Physik 38 (1931). Turing, Alan. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: Proceedings of the London Mathematical Society 2, no. 42 (1936).

Nevertheless, both aspects of Bense's hypothetical project became reality over time, and in the late 1990s, two artistic sub-genres had established themselves, both with a huge theoretical framework attached to them: net literature/poetry, which was mainly concerned with the possibilities of the hypertext[7] and looked at computer code only in terms of "style" (basically: lots of brackets and ellipses), and software art, with a theoretical discourse focused on the performativity of actual computer code.[8] Both sub-genres overlapped to some degree, but not too much,[9] with a huge part of the actual artistic content being much more geared towards making a theoretical point, rather than towards exploring the aesthetic properties of the medium. Finally, somewhere around the mid-2000s, with consumer grade computers becoming powerful enough, the whole focus shifted from text to images. Net literature/poetry died a quiet death and was buried together with the whole net art genre somewhere in the more shady corners of the World Wide Web.

But what happened to software art? Why did it lose its right to exist as well, both as a term and an independent sub-genre of media art? The reasons for this are, of course, manifold. One particular reason, however, deserves our attention: The accidental education of the artist in the domain of computer programming. With the Processing project winning a Golden Nica[10] precisely in 2005, programming knowledge – at least on the level of tools geared towards artists, such as Processing – had become so ubiquitous that the "software art" specifier was not needed anymore. This does not mean, however, that the "expert" artist, the artist that had achieved a level of proficiency in the domain of computer programming equal to that of an actual software engineer, disappeared as well. Instead, the whole scale bit-shifted to the left:[11] artists with no knowledge about computer programming became artists with some knowledge about computer programming (for instance within the limits of an environment like Processing), artists with some knowledge about computer programming became artists with a lot of knowledge about computer programming.

[7] See, for instance: Suter, Beat, and Michael Böhler (eds.). *Hyperfiction. Hyperliterarisches Lesebuch: Internet und Literatur.* Basel: Stroemfeld/Nexus, 1999.

[8] See, for instance: Arns, Inke. "READ_ME, RUN_ME, EXECUTE_ME: SOFTWARE AND ITS DISCONTENTS, OR : IT'S THE PERFORMATIVITY OF CODE, STUPID". http://art.runme.org/1107863582-4268-0/arns.pdf

[9] Cramer, Florian: "Warum es zuwenig interessante Netzdichtung gibt. Neun Thesen". http://www.netzliteratur.net/cramer/karlsruher_thesen.html

[10] The Golden Nica is the highest prize of the Ars Electronica Festival in Linz, Austria, and one of the most important awards in media arts.

[11] In computer programming, bit shifting is a useful bitwise operation: the bits in a variable are shifted (in this case to the left) and the freed up least-significant bit set to zero. The method can be used, for instance, to create very large numbers without writing them out in the source code. Bit shifting is also a good example for the type of possible algorithmic simplification that is the basis of the concept of Kolmogorov complexity, see further down.

To look at it again from a historical perspective: the “expert” artists of the 1960s were experts at writing programs that, due to the evolution of computer technology, today are more than easy to reproduce for anyone with a consumer grade computer and an Internet connection. Then, in the 1990s, the scale shifted radically for the first time. More artists started to use easy scripting languages, like HTML, while one of the expert artists of that time, Jeffrey Shaw, already worked with complex computer graphics programming tools built in C.[12] Finally, today, the creation of 3D generative content is not an expert domain anymore, and the expert artists, like Andy Lomas,[13] use high-performance computing tools on the GPU for the creation of their works. Many other comparisons of this sort can be inferred if we look closely enough at the state of media art today.

As it is the vertical[14] nature of paradigm shifts, the further evolution of this historically significant process of accidental education can not be foretold. Its mere existence, however, is reason enough to re-think the role of source code in the exhibition of media art. What does this entail? Certainly not the resurrection of software art, even less the reactivation of Max Bense’s structuralist utopia that is still basking in the fluorescent light of the most efficient algorithm. What it does entail is something entirely pragmatic: basic knowledge of computer programming is not expert knowledge, and computer code not a secret language anymore. Instead of treating it as an auratic, non-specific signifier for the impenetrable complexity of technology, like it was and still is common even for contemporary exhibitions, computer code should be treated as a commentary that provides an additional layer of meaning, not only adding to the aesthetic complexity of the work but also making more explicit its relationship to the relationships of production by exposing the tools it was made with.

[12] For the seminal work *The Legible City* (1989-1991). http://www.jeffrey-shaw.net/html_main/show_work.php?record_id=83

[13] Lomas, Andy. “Cellular Forms: an Artistic Exploration of Morphogenesis”. ACM SIGGRAPH 2014 Studio.

[14] See Derrida, Jacques: Derrida, Jacques. “A Certain Impossible Possibility of Saying the Event”. In: Critical Inquiry 33(2), 2007.

Finally, treating computer code as a commentary, rather than, as the notion of software art suggested, a work in itself, makes peace with the central problem of using computer code within an artistic context: its inability to be ambiguous, to convey more than one meaning at the same time.

That this inability does in fact exist, is easily proven if we look at an algorithmic definition of information, like Kolmogorov complexity.[15] The premise of Kolmogorov complexity is simple: the measure of "information" for a string of characters is defined as the shortest algorithm implemented in any Turing-complete language to reproduce this string. For instance, imagine a string $\boldsymbol{x}_1$ with 500 different characters and a string $\boldsymbol{x}_2$ with 500 times the same character. The size of the algorithm $\mathbf{K}$ to reproduce $\boldsymbol{x}_1$ is at least the size of the string itself. $\mathbf{K}(\boldsymbol{x}_2)$, however, could be less than $\boldsymbol{x}_2$, as we can imagine an algorithm in an optimal language $\mathbf{L}$ that just prints one character 500 times. The implication important for our argument here is: the language in which the algorithm is implemented does not matter. This becomes apparent if we imagine a program in the optimal language $\mathbf{L}$ for translating a program from any language into $\mathbf{L}$. For any algorithmic reconstruction of a string $\boldsymbol{y}$ in any Turing-complete language $\mathbf{N}$ the upper bound here is $\mathbf{K}(\boldsymbol{y}) + \mathbf{K}(\mathbf{N}{\rightarrow}\mathbf{L})$, with $\mathbf{K}(\mathbf{N}{\rightarrow}\mathbf{L})$ being a constant that does not depend on $\boldsymbol{y}$. Hence, the apparent "complexity" of any programming language is negligible in terms of the actual complexity of the algorithm implemented. Compared

[15] Kolmogorov, Andrei N: "Three Approaches to the Quantitative Definition of Information". In: Problemy Peredachi Informatsii 1/1 (1965).

to natural language — not to speak of "poetic" language — that is inherently "overloaded"[16] with meaning, programming languages are inherently redundant and just an arbitrary expression of the same Turing-complete functions. Their "style" is purely visual.[17] This redundancy is a necessary prerequisite to their performative power. In a way, computer code, to the aesthetic analysis, is always in a state of superposition. We can not look at it without altering it, and we have to choose our perspective wisely. If we treat computer code as *text*, we can not treat it as an *algorithm*, and vice versa.

By showcasing source code as a legible, non-auratic commentary to the aesthetic artifacts — and precisely not a matter of the heart — both on the following pages as well as in the exhibition, OPEN SOURCES, as one of the first exhibitions to do so, tries to take this realization seriously.

[16] In programming, overloading a function means to define it more than once to be able to call it with different parameters. No programming language allows to have the exact same definition more than once though, which is precisely what natural language does, with the actual meaning not being completely "resolvable" but continuously oscillating between its poles. See also Benjamin, Walter: "Die Aufgabe des Übersetzers". In: *Gesammelte Schriften Bd. IV/1*, p. 9-21. Frankfurt/Main: Suhrkamp, 1972. The title itself translates either to: "the translator's task" or "the translator giving-up".

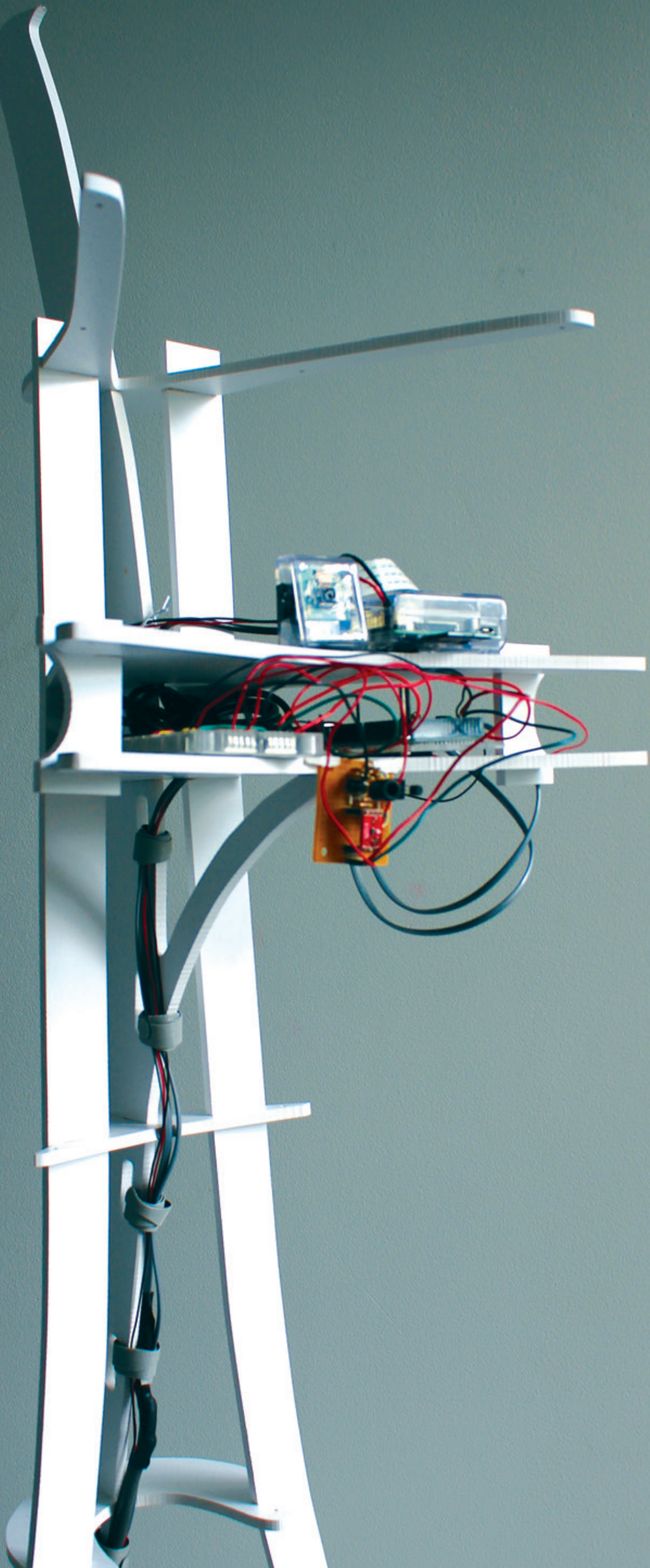
[17] This does, of course, not mean that a purely visual analysis could not yield interesting results. See: Hagen, Wolfgang. "The Style of Sources. Remarks on the Theory and History of Programming Languages". http://www.whagen.de/publications/2005/DigitalMediaRoutledge/RT2241_C0091.pdf

Kodama: Me the Tree

TIM **WOOD**

Kodama is an exploration in the interface between alternate worlds: the physical, the virtual, and the imaginary. This iteration focuses on the interaction of various virtual tree structures/simulations with the human body. This project ideally would live outside, projected at a large scale. Through interaction with Kinect-based motion tracking, I want to magnify/accentuate the shape and movement of participants in combination with virtual tree forms growing on and around the body/silhouette.

```
/*
 * SeerScript (scala 2.11.2)
 *
 * Space colonizing tree growth onto user depth image
 *
 *   put this code into:
 *    https://github.com/fishuyo/seer/tree/devel/
examples/scripts/live.scala
 *
 * 2015 Tim Wood
 */

import com.fishuyo.seer.openni._
import com.fishuyo.seer.particle._
import com.fishuyo.seer.audio._

import collection.mutable.ArrayBuffer
import collection.mutable.ListBuffer
import collection.mutable.HashSet


class Branch(var parent:Branch, var pos:Vec3, var
growDirection:Vec3){
 var growDirection0 = Vec3(growDirection)
 var growCount = 0
 var age = 0

 def grow(){
  age += 1
  if(parent != null) parent.grow()
 }

 def reset(){
  growCount = 0
  growDirection = growDirection0
 }
}

class Leaf(var pos:Vec3){
 var closest:Branch = null
}

class Tree {
 var done = false
 var pos = Vec3()

 var minDistance = 0.05
 var maxDistance = 0.1 //0.35
 var branchLength = 0.04

 var root = new Branch(null, Vec3(), Vec3(0,1,0))
 var leaves = ListBuffer[Leaf]()

 var branches = Octree[Branch](Vec3(0),5)

 branches += (root.pos -> root)

 def reset(){
  branches.clear
  leaves.clear
  branches += (root.pos -> root)
 }

 def grow(){

  if (leaves.size == 0) {
    return
  }

  //process the leaves
  var i = 0
  while( i < leaves.size){

   var leafRemoved = false

   var direction = Vec3()
   val leaf = leaves(i)
   leaf.closest = null

   //Find the nearest branch for this leaf
   var break = false
   val near = branches.getInSphere(leaf.pos, maxDistance)
   near.values.foreach( (b) => {
    if(!break){
     direction = leaf.pos - b.pos
     val dist = direction.mag
     direction.normalize

     if( dist <= minDistance){
      leaves -= leaf
      i -= 1
      leafRemoved = true
      break = true
     } else if( dist <= maxDistance){
      if( leaf.closest == null)
       leaf.closest = b
      else if ( (leaf.pos - leaf.closest.pos).mag > dist)
       leaf.closest = b
     }
    }
   })
   //if the leaf was removed, skip
   if (!leafRemoved){
     //Set the grow parameters on all the closest branches
that are in range
     if (leaf.closest != null){
       val dir = leaf.pos - leaf.closest.pos
       dir.normalize()
       leaf.closest.growDirection += dir      //add to grow
direction of branch
       leaf.closest.growCount += 1
     }
   }

   i += 1
  }

  //Generate the new branches
  val newBranches = HashSet[Branch]()
  branches.foreach( (p,b) => {
   if (b.growCount > 0){    //if at least one leaf is affecting
the branch

     val avgDirection = b.growDirection / b.growCount
     avgDirection.normalize()

     val newBranch = new Branch(b, b.pos + avgDirection
* branchLength, avgDirection);
     b.grow()

     newBranches += newBranch
     b.reset()
   }
  })

  //Add the new branches to the tree
  var branchAdded = false;
  newBranches.foreach( (b) => {
    branches += (b.pos -> b)
    branchAdded = true
  })
 }
}

object Script extends SeerScript {

 var inited = false

 OpenNI.initAll()
 OpenNI.start()
 OpenNI.pointCloud = true
 OpenNI.pointCloudDensity = 2

 val skeleton = OpenNI.getSkeleton(1)

 val mesh = new Mesh()
 mesh.primitive = Points
 mesh.maxVertices = 640*480
 mesh.maxIndices = 10000
 val model = Model(mesh)
 model.material = Material.basic
 model.material.color = RGBA(1,1,1,1)
 model.material.transparent = true

 var grow = false
 val tree = new Tree

 tree.minDistance = 0.05
 tree.maxDistance = 0.1
 tree.branchLength = 0.04

 val treeMesh = Mesh()
 treeMesh.primitive = Points
 treeMesh.maxVertices = 100000
 val treeModel = Model(treeMesh)
 treeModel.material = Material.basic
 treeModel.material.color = RGBA(1,1,1,1)

 var write = false

 Keyboard.clear
 Keyboard.use
 Keyboard.bind(“g”, () => { grow = !grow })
 Keyboard.bind(“r”, () => { tree.reset;  })
 Keyboard.bind(“p”, () => { write = true })


 override def init(){
  inited = true
  Camera.nav.pos.set(0f,0f,-0.8334836f)
  tree.root.pos.set(0,0.1,-1.6)
 }

 override def draw(){
  FPS.print

  model.draw

  treeMesh.clear()
  tree.branches.getAll.values.foreach( (b) => {
   if(b.parent != null){
    // drawBranchRect(treeMesh, b)
    drawBranchRing(treeMesh, b)
   }
  })
  treeMesh.update()
  treeModel.draw()

  if(write){
   treeMesh.writePointCloud()
   mesh.writePointCloud()
   write = false
  }
 }

 def drawBranchRect(m:Mesh, b:Branch){
  val s = Vec3(0.00001,0,0)
  m.vertices += b.pos - s * b.age
  m.vertices += b.parent.pos - s * b.parent.age
  m.vertices += b.parent.pos - s * b.parent.age
  m.vertices += b.parent.pos + s * b.parent.age
  m.vertices += b.parent.pos + s * b.parent.age
  m.vertices += b.pos + s * b.age
  m.vertices += b.pos + s * b.age
  m.vertices += b.pos - s * b.age
 }

 def drawBranchRing(m:Mesh, b:Branch, r:Float =
0.00001f){
  var n = 1
  if(b.age < 50) n = 3
  else if(b.age < 100) n = 4
  else if(b.age < 150) n = 5
  else if(b.age < 400) n = 6
  else if(b.age < 600) n = 10
  else n = 20

  for( i <- 0 until n){
   val phase = i.toFloat / n * 2 * Pi
   val cos = r*math.cos(phase)
   val sin = r*math.sin(phase)
   val vx = Vec3(1,0,0)
   val vz = Vec3(0,0,1)
   val off1 = vx * cos * b.age + vz * sin * b.age
   if(off1.mag < 0.005) off1.set(off1.normalized * 0.005)
   m.vertices += b.pos + off1
   m.normals += off1.normalized
  }
 }

 override def animate(dt:Float){
  if(!inited) init()

  try{
   skeleton.updateJoints()
   // println(skeleton.joints(“head”))

   mesh.clear
   mesh.vertices ++= OpenNI.pointMesh.vertices
   mesh.update

   tree.leaves.clear
   tree.leaves ++= OpenNI.pointMesh.vertices.map( new
Leaf(_) )
   if(grow) tree.grow()

  } catch { case e:Exception => println(e) }
 }
}

Script
```



R.O.V.E.R. 3.0

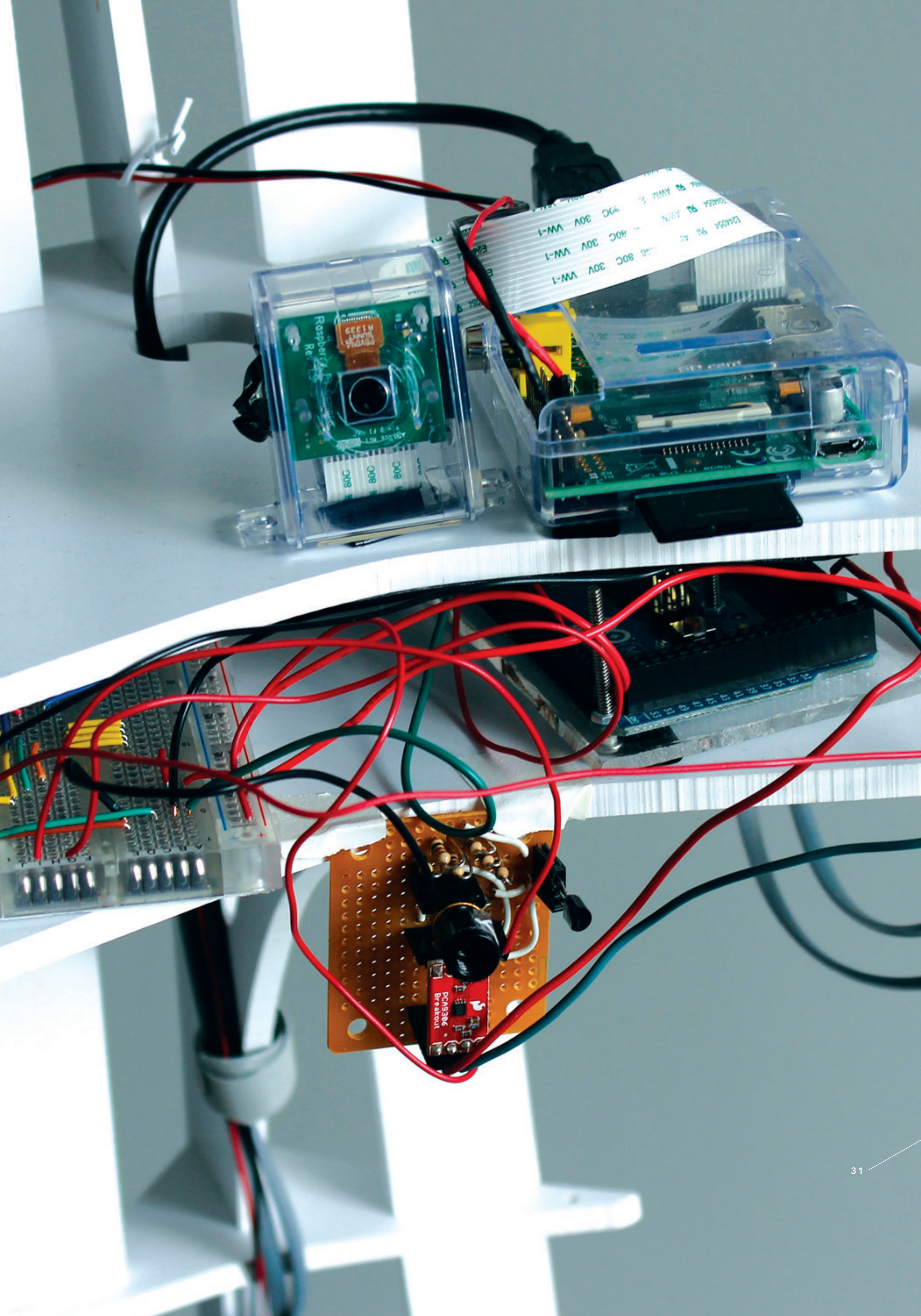
HANNAH **WOLFE**

ROVER, the Reactive Observant Vacuous emotive robot, is a modular learning platform to explore human robot interaction. This cross disciplinary project explores computer vision, human computer interaction (HCI), machine learning, sculpture, audio, physical computing, interactive environments and bio-mimicry. *ROVER* can navigate a given space, avoid obstacles, find people and attempt to communicate with them. *ROVER* takes human interaction as feedback for a genetic algorithm to create emotive sound. The robot uses accessible technology including the Arduino, Raspberry Pi, iRobot Create, and infrared sensors. After this iteration is complete this project can be used as a platform for further investigation into human robot interaction by changing the program.

```
#include <Roomba.h>
#include <i2cmaster.h>
#include <DistanceGP2Y0A21YK.h>

//variables for distance sensors
DistanceGP2Y0A21YK DistForward;
DistanceGP2Y0A21YK DistBackward;
DistanceGP2Y0A21YK DistLeft;
DistanceGP2Y0A21YK DistRight;

// Defines the Roomba instance and the HardwareSerial it
connected to. It also begins the serial.
Roomba roomba(&Serial3);
byte dataBuffer[20];
bool connected = false;

// Variables for roomba movement
bool lastTurnLeft = true;

// Variables for the Thermosensor
int freq = 16;  //Set this value to your desired refresh
frequency
int IRDATA[64];
byte CFG_LSB, CFG_MSB, PTAT_LSB, PTAT_MSB, CPIX_LSB,
CPIX_MSB, PIX_LSB, PIX_MSB;
int PIX, v_th, CPIX;
float ta, to, emissivity, k_t1, k_t2;
float temperatures[64];
int count=0;
unsigned int PTAT;
int a_cp, b_cp, tgc, b_i_scale;

int a_ij[64];
int b_ij[64];
float alpha_ij[64];
float v_ir_tgc_comp[64];

#include “MLX90620.h”

unsigned long lastPollTime = 0;

void Temperatures_Serial_Transmit(){
  for(int i=0;i<=63;i++){  //this should go up to 63
  //  if(!(i % 16)) Serial.println();
    Serial.print(temperatures[i]);
    Serial.print(“|”);
  }
  Serial.write(“::”);
}


bool checkSong(){
  if(Serial.available() > 0) {
    return true;
  }
  return false;
}

void calibrate_MLX90620() {
  read_PTAT_Reg_MLX90620(); // ambient temperature
  calculate_TA(); // calculate absolute chip temperature
  check_Config_Reg_MLX90620();
}

void read_MLX90620() {
  if (count == 0)
    calibrate_MLX90620();

  read_IR_ALL_MLX90620();  //should be called more
frequently
  read_CPIX_Reg_MLX90620();
  calculate_TO();

  ++count;
  if (count == 16)
    count = 0;
}

void bootup() {
  connected = false;
  while (!connected) {
    roomba.start();
    roomba.fullMode();
    roomba.drive(0, 0);

 //   Serial.println(“Waiting startup msg”);

    Serial3.write(142);
    Serial3.write(Roomba::SensorBumpsAndWheelDrops);

    delay(200);
    if (Serial3.available()) {
      Serial3.read();
      if (!Serial3.available()) {
        connected = true;
      }
    }

    while(Serial3.available())
      Serial3.read();
  }
}
```

```
void setup() {
  Serial.begin(57600);
  Serial3.begin(57600);
  delay(500);

  DistForward.begin(A0);
  DistBackward.begin(A1);
  DistLeft.begin(A3);
  DistRight.begin(A2);

  bootup();

 // Serial.println(“Roomba Initialized”);

  i2c_init();
  PORTC = (1 << PORTC4) | (1 << PORTC5);
  delay(5);
  read_EEPROM_MLX90620();
  delay(5);
  config_MLX90620_Hz(freq);
  delay(5);
}

void DistanceMovement()
{
  int dF = DistForward.getDistanceCentimeter();
  int dB = DistBackward.getDistanceCentimeter();
  int dL = DistLeft.getDistanceCentimeter();
  int dR = DistRight.getDistanceCentimeter();
  if (dL < 7)
  {
    if (dB < 7)
    {
      drive(50, 10);
    }
    else
    {
      drive(10, -100);
      lastTurnLeft = true;
    }
  }
  else if (dR < 7)
  {
    if (dB < 7)
    {
      drive(10, 50);
    }
    else
    {
      drive(-100, 10);
      lastTurnLeft = false;
    }
  }
  else
  {
    if (dF < 7)
    {
      if (!lastTurnLeft)
        drive(-100, 10);
      else drive(10, -100);
    }
    else
    {
      HeatMovement();
    }
  }
}

void HeatMovement()
{
  int maxValue = -1;
  int i = -1;
  for (int j = 0; j < 64; ++j)
  {
    if(temperatures[j] > maxValue && temperatures[j] <
40.0)
    {
      maxValue = temperatures[j];
      i = j;
    }
  }
  if (i % 16 >= 0 && i % 16 < 7)
  {
    drive(50, 10);
  }
  else if (i % 16 > 8)
  {
    drive(10, 50);
  }
  else if (i % 16 == 7||i % 16 ==8)
  {
    drive(50, 50);
  }
  else
  {
    if (lastTurnLeft){
      drive(50,-10);
    }
    else drive(-10,50);
  }
}

void drive(int l, int r)
```

```
{
  roomba.driveDirect(l, r);
  Serial.println(r);
  Serial.println(l);
}

void loop() {
  bool songReceived = checkSong();
  if (songReceived)
  {
    drive(0,0);
    delay(5000);
  }
  /// movement


  uint8_t sensors[3] =
{Roomba::SensorBumpsAndWheelDrops,
Roomba::SensorAngle, Roomba::SensorDistance};

  bool validSensor = roomba.getSensorsList(sensors,
sizeof(sensors), dataBuffer, 5);
  if (validSensor) {
  //  Serial.print(“ got sensor data: “);
  //  Serial.println(dataBuffer[0], BIN);
    int angle = dataBuffer[2] | dataBuffer[1] << 8;
    int distance = dataBuffer[4] | dataBuffer[3] << 8;
 //   int angle = dataBuffer[2] << 8 | dataBuffer[1];
   // int distance = dataBuffer[4] << 8 | dataBuffer[3];
    Serial.println(angle);
    Serial.println(distance);

    if ((dataBuffer[0] & 0xE0) != 0x00) {
   //   Serial.println(“Lost Connection”);
      bootup();
    }
    if ((dataBuffer[0] & 0x03) == 0x00) {
      //here we want to check the distance sensors
      DistanceMovement();
     // HeatMovement();
    }
    else if ((dataBuffer[0] & 0x03) == 0x01) {
      drive(-100, 10);
      lastTurnLeft = false;
    }
    else if ((dataBuffer[0] & 0x03) == 0x02) {
      drive(10, -100);
      lastTurnLeft = true;
    }
    else if ((dataBuffer[0] & 0x03) == 0x03) {
      if (!lastTurnLeft)
        drive(-100, 10);
      else drive(10, -100);
    }
  }
  else {
  //  Serial.println(“Lost Connection”);
    bootup();
  }
  delay(100);
  /// end movement

  if (millis() - lastPollTime > 125) {
    read_MLX90620();
    lastPollTime = millis();
    Temperatures_Serial_Transmit();
  }

}
```



C1
D1
E1
F1
G1
A1
B1
C2
D2
E2
F2
G2
A2
ADVANCED FUNCTIONS
VOLUME
MAX
MIN
OCTAVE
MAX
MIN
PITCH BEND
MAX
MIN
MODULATION

Color Music

DAVID **GORDON**

Color Music is an installation investigating multimodal perception and the live generation of music. The installation consists of a software instrument for creating live algorithmic music and visuals, which the user performs using a colored keyboard and a MIDI controller.

The generative system consists of one or more sonifying agents moving through a virtual space filled with two overlapping 3D grids of objects: a rhythmic grid representing beats and a harmonic grid representing a network of chords in thirds and fifths, or *Tonnetz*. Cells on the rhythmic grid represent notes; each note's color represents its pitch, shape represents length, size represents volume, and the absence of a note represents a rest. Cells on the harmonic grid represent up to seven possible chords within a scale. The user can input note cells and harmonic cells into the system using the keyboard. Whenever one of the sonifying agents collides with a note, that note is played. When an agent collides with a harmonic cell, it alters the harmony, changing the other notes around it. The combined collisions of all the agents generate the structure of a musical performance.

The user can control many parameters of the system in real-time such as tonal center, scale mode, tempo, harmonic rhythm and the number of agents affecting each grid.

```
/**************************/
/*  From Melody3D Class  */
/**************************/

void generateMelody()
{
  float pos = 0.;
  int count = 0;
  int notePitch = 0;
  boolean nextToLast = false, lastNote = false;

  newNotes = new ArrayList<MidiNote>();

  while(pos<duration-1)
  {
    int randMax = int(1./minNoteDuration)+1;
    int randNum = int(random(1, randMax));
    float noteDur = constrain(randNum*minNoteDuration,
minNoteDuration, 1.);

    float noiseVal = noise(pos * noiseScale);
    notePitch = int(map(noise(pos*noiseScale), 0., 1., 0, 11));

    while(!noteIsInScale(notePitch, mode))
    {
       notePitch += 1;
    }

    if(pos + noteDur > duration - 1. && !lastNote)    // Second to last note
    {
      noteDur = duration - 1. - pos;                                        // Last note
must be a full beat
      nextToLast = true;
      count++;
    }
    else if(pos + noteDur > duration - 1. && lastNote)  // Last note
    {
       newNotes.add( new MidiNote( notePitch + 60, 3, frameCount+int(pos
* 30.), noteDur, 100, count ) );
       count++;
    }
    else                                        // All other notes
    {
      newNotes.add( new MidiNote( notePitch, 3, int(pos * 30.), noteDur,
100, count ) );
      pos += noteDur;
      count ++;
    }

    if(nextToLast) lastNote = true;
  }

  melodyLength = count + 1;
  notes = newNotes;
}

void createRandomMelody()                                            // Generate
new random melody
{
 int tonic = int(random(0,12));
 melody = new Melody(tonic, melodyLength, minimumDuration);
 melody.generateMelody();
}

void createMelodyInCurrentKey()                                      // Generate
new melody in current key
{
 melody = new Melody(findPentatonic(0,7), melodyLength,
minimumDuration);
 melody.generateMelody();
}

void arpeggiateNoteUp(int pitch)
{
   arpeggio = new Melody(pitch, 10, 0.5/4);
   arpeggio.arpeggiateChord(true);
   arpeggio.isPlaying = true;
}

void arpeggiateNoteDown(int pitch)
{
   arpeggio = new Melody(pitch, 10, 0.5/4);
   arpeggio.arpeggiateChord(false);
   arpeggio.isPlaying = true;
}
```

```
void play()
{
  for (int i=0; i < notes.size(); i++)
  {
    MidiNote note = notes.get(i);
    int pos = frameCount % melodyLength;      // USE THIS instead of framecount

    if(note.time <= frameCount && note.time + note.duration*30 > frameCount &&
note.isPlaying == false)
    {
      notesToPlay.add(note);
    }

    if(notesToPlay.size() > 0 && notesPlaying.size() < 5)       // Avoids sending too many
MIDI messages
    {
       note = notesToPlay.get(0);

       if( frameCount % notesToPlay.size()== i )
       {
         playMidiNote(note.pitch + 12 * note.octave, note.duration, note.velocity);
         curNote = note.noteID;
         notesToPlay.remove(note);
       }

       note.isPlaying = true;
    }

    if(note.isPlaying && note.time + note.duration < curTime)
      note.isPlaying = false;
  }
}
```

```
/************************/
/*  From Note3D Class  */
/************************/

color setColorFromPitch()
 switch(currentPitch)
 {
   case 0:
         r = 200-hue/2.5;
         g = hue/3+50;
         b = hue;
         a = map(pit, 0, sqrt(127), 0, alpha*map(n.getVelocity(), 0, 50, 0.6, 1));
    break;

   case 1:
         r = hue/2.5;
         g = 200;
         b = hue * 1.3;
         a = map(pit, 0, sqrt(127), 0, alpha*map(n.getVelocity(), 0, 50, 0.6, 1));
    break;

   case 2:
         r = hue/1.5;
         g = 0;
         b = hue * 1.;
         a = map(pit, 0, sqrt(127), 0, alpha*map(n.getVelocity(), 0, 50, 0.6, 1));
    break;
   case 3:
         r = 0;
         g = 12 / hue;
         b = hue * 1.;
         a = map(pit, 0, sqrt(127), 0, alpha*map(n.getVelocity(), 0, 50, 0.6, 1));
    break;
   case 4:
         r = 20/hue;
         g = hue * 1.5;
         b = hue * 2.;
         a = map(pit, 0, sqrt(127), 0, alpha*map(n.getVelocity(), 0, 50, 0.6, 1));
    break;
     default:
          r = 115;
         g = 200-hue/3;
         b = hue * 1.3;
         a = map(pit, 0, sqrt(127), 0, alpha*map(n.getVelocity(), 0, 50, 0.6, 1));
    break;
 }

  fill(r, g, b, a);
  stroke(hue,pit);
}
```

A conversation with

Matthew **Wright**

The EoYS team

A program like ours, at the intersection of arts and engineering, does it create a specific challenge to balance both sides?

It’s extremely challenging. Incoming students span the spectrum from expert to knowing nothing about various aspects of arts and engineering, and negotiating that is one of the most difficult aspects of teaching in MAT.

I think not in terms of a “middle ground” but in terms of people having the union (in set theory terms) of the skills, knowledge, outlook, etc., of three or four disciplines.

When all these technologies being explored at MAT become part of the regular toolset of many disciplines, will programs like MAT also merge into other disciplines?

I doubt it.

Other disciplines already use a lot of the same technologies as we – for example –, working with materials scientists through the AlloSphere, they know much more about voxels and meshes and rendering than I did (coming from a computer music background). But just because they know these technologies doesn't mean they're able or willing to make transdisciplinary art with them.

As part of academia, generations of students come and go, but faculty stays. Do you see any difference between students from, let's say, ten years ago and the current generation of students?

10 years ago I was a grad student in a computer music center; it's hard for me to make this comparison. Obviously certainly today's students have "newer" technological knowledge.

The topic of our exhibition this year is **OPEN SOURCES.** We will showcase not only the works themselves but also excerpts from their source code, to make an exhibition that is more open, more interesting, and more challenging to a technically educated public. What role does coding play in your work?

A huge one. It's the basis of almost all my work and teaching.

Are you "language agnostic"?

No, I believe that different tools are more and less appropriate for various different jobs.

To me the interesting question is an institution's choice to impose a certain set of tools or not. CNMAT / UC Berkeley is totally a *Max/MSP/ Jitter* shop. CCRMA / Stanford has their own "Planet CCRMA" toolbox but there's a lot of diversity in what students and faculty use for their own projects. MAT is totally all over the place, with almost every class and every student using a different programming language.

Do you agree with the very common sentiment that everybody should learn to code?

No, but I believe learning to code is very empowering for people who use computers on a daily basis. That's a minority of the earth's population and maybe it should be a smaller one.

Many of your activities in MAT and in the Music department involve performance and interpretation. Is there a reason why performance has taken a predominant role over composition?

I don't see performance as in any way opposed to composition. Much of my musical work involves improvisation, which I consider a form of composition.

The open sound control (OSC) protocol is now widely the standard for communication in the electronic music domain.
Could you tell us a bit more how it came to be?

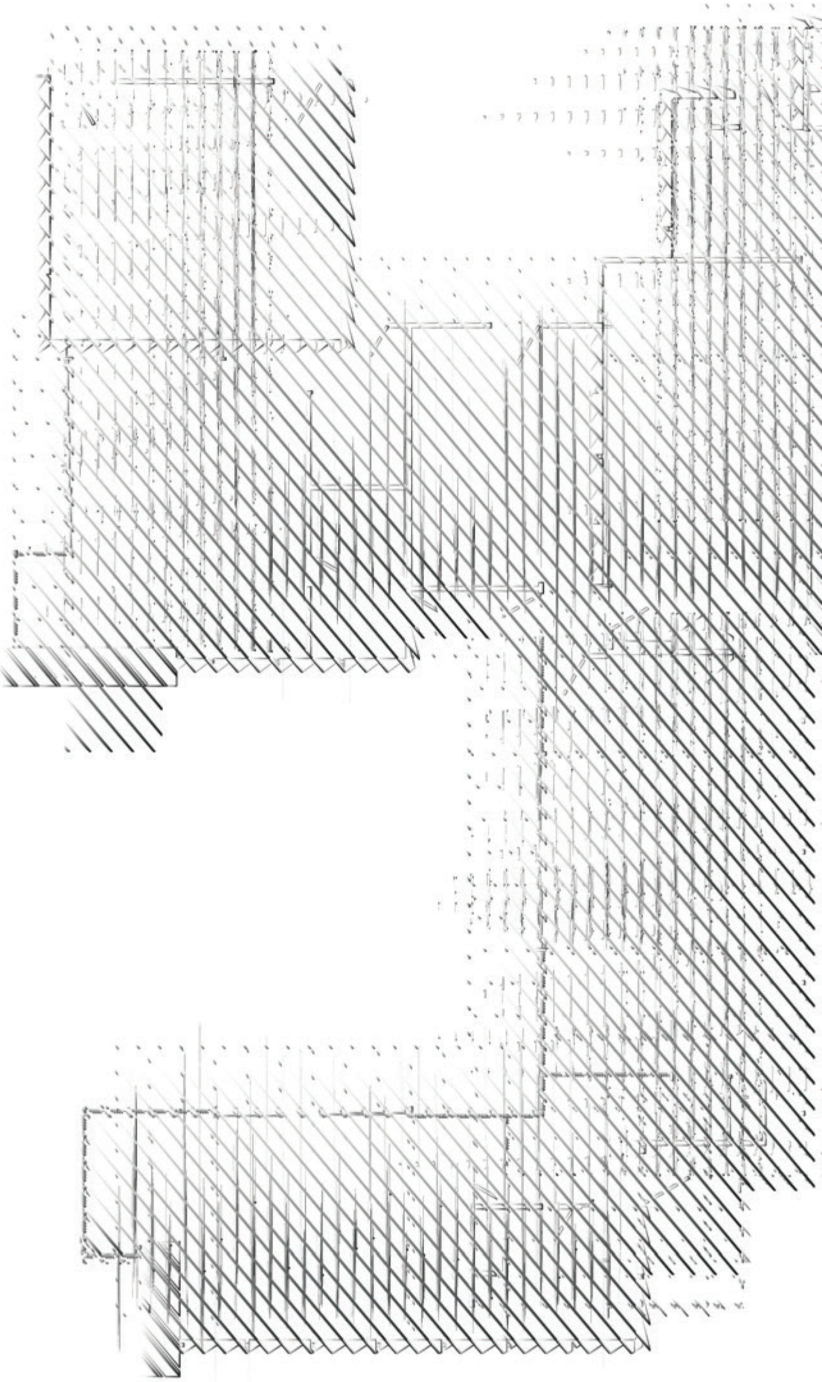
Long story. Here's the beginning:

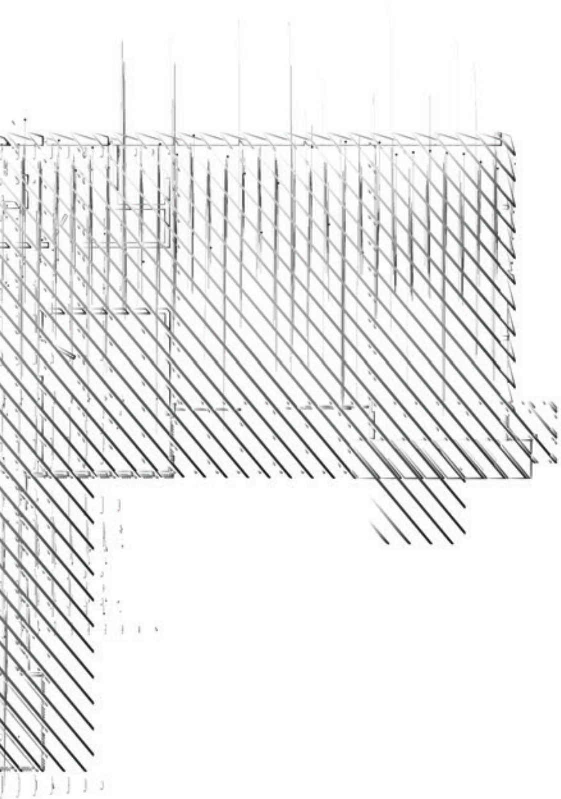
OSC started as a generalization of a custom networking protocol allowing additive synthesis running on one computer platform (Custom *C* synthesizer running on *SGI*) to be controlled in real time from another platform (custom *Max* patch running on Macintosh) integrating gestural inputs with interactive control structures via then-modern 10-megabit per second Ethernet. (In 2015, 100M ethernet is slow, 1G is standard, and 10G is fast.)

Although MIDI was ubiquitous (and indeed part of the full system, bringing interactive human control inputs into the *Max* patch in real time), its built-in addressing structure of 16 channels each a 128-note keyboard and the lack of a representation for floating point numbers could not be mapped to the affordances of this particular synthesizer. Instead this synth had 26 voices, each of which could be set to a "timbral prototype". (an already existing sinusoidal model specifying frequencies and amplitudes over time for some number of oscillators.)

Any given "virtual time" would result in a specific spectrum taken (via interpolation) from the sinusoidal model, and so each voice had a "time machine" controlling the progression of virtual time over real time, allowing commands such as "go to the 7 second point then advance backwards at 50% speed."

Also there were a series of possible spectral modifications, such as adjusting the relative balance of even- and odd-numbered harmonics, perturbing frequencies to alter harmonicity, etc. We designed a protocol "SynthControl" only to be able to control the features of this particular synthesizer, using the letters A-Z to name the 26 voices and with symbolic names.



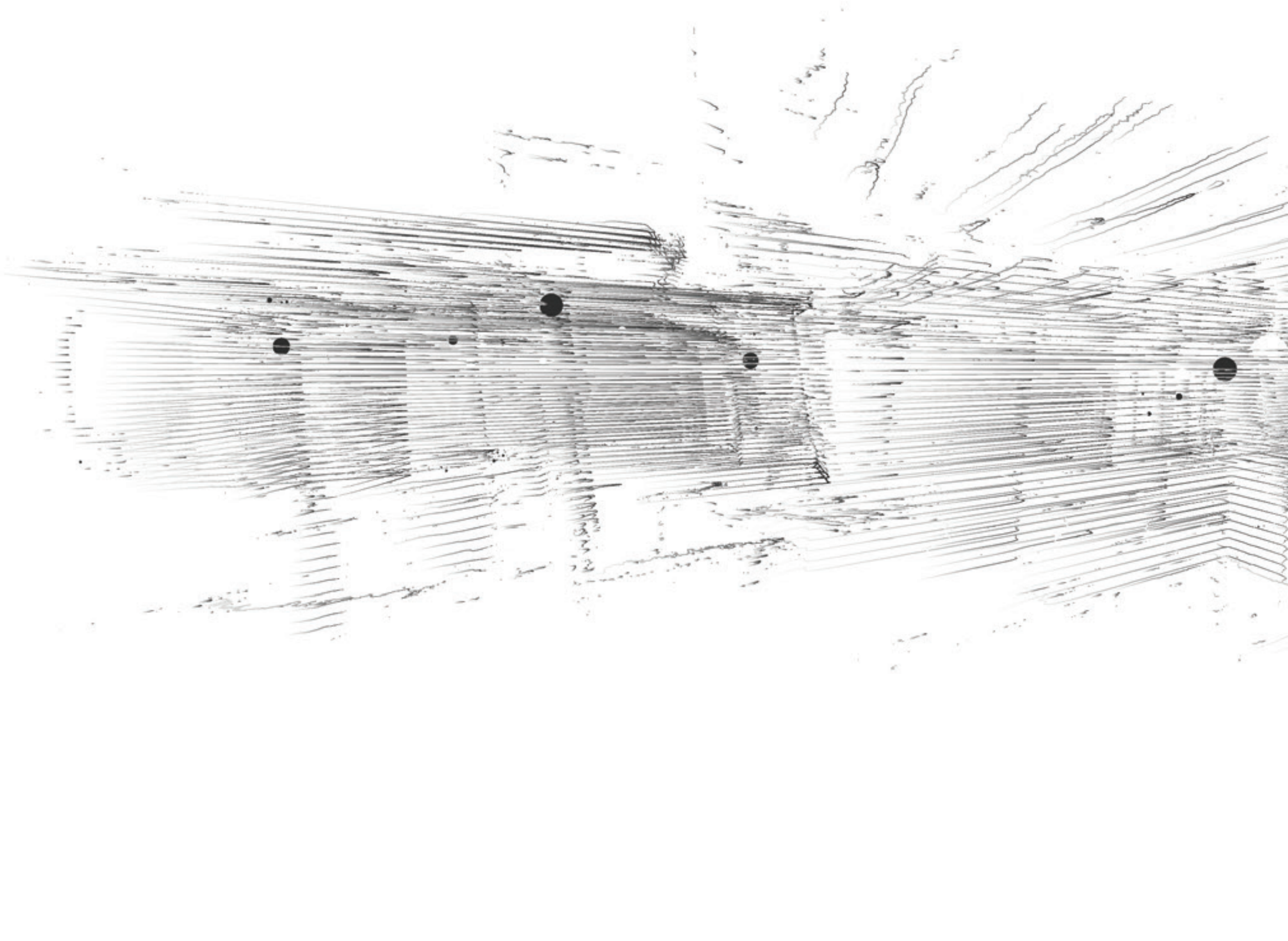


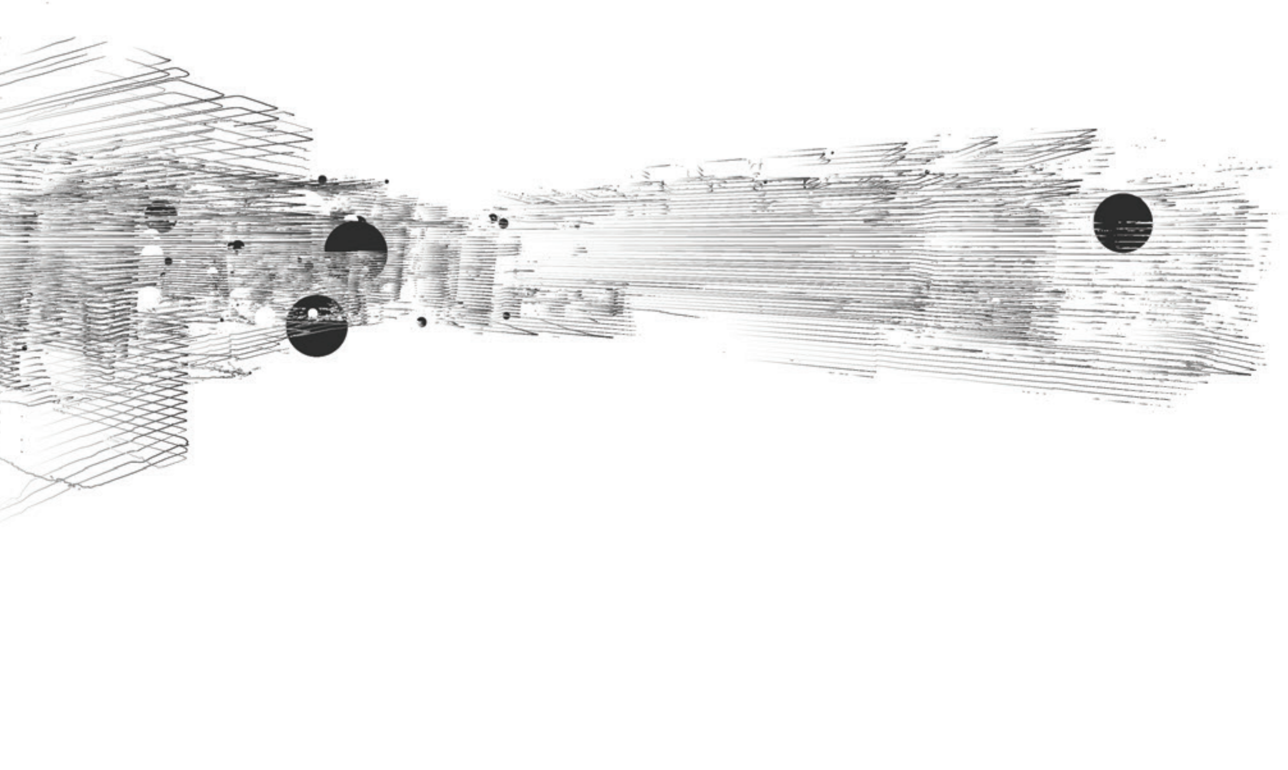
SoundScan: Sound and the Schindler House

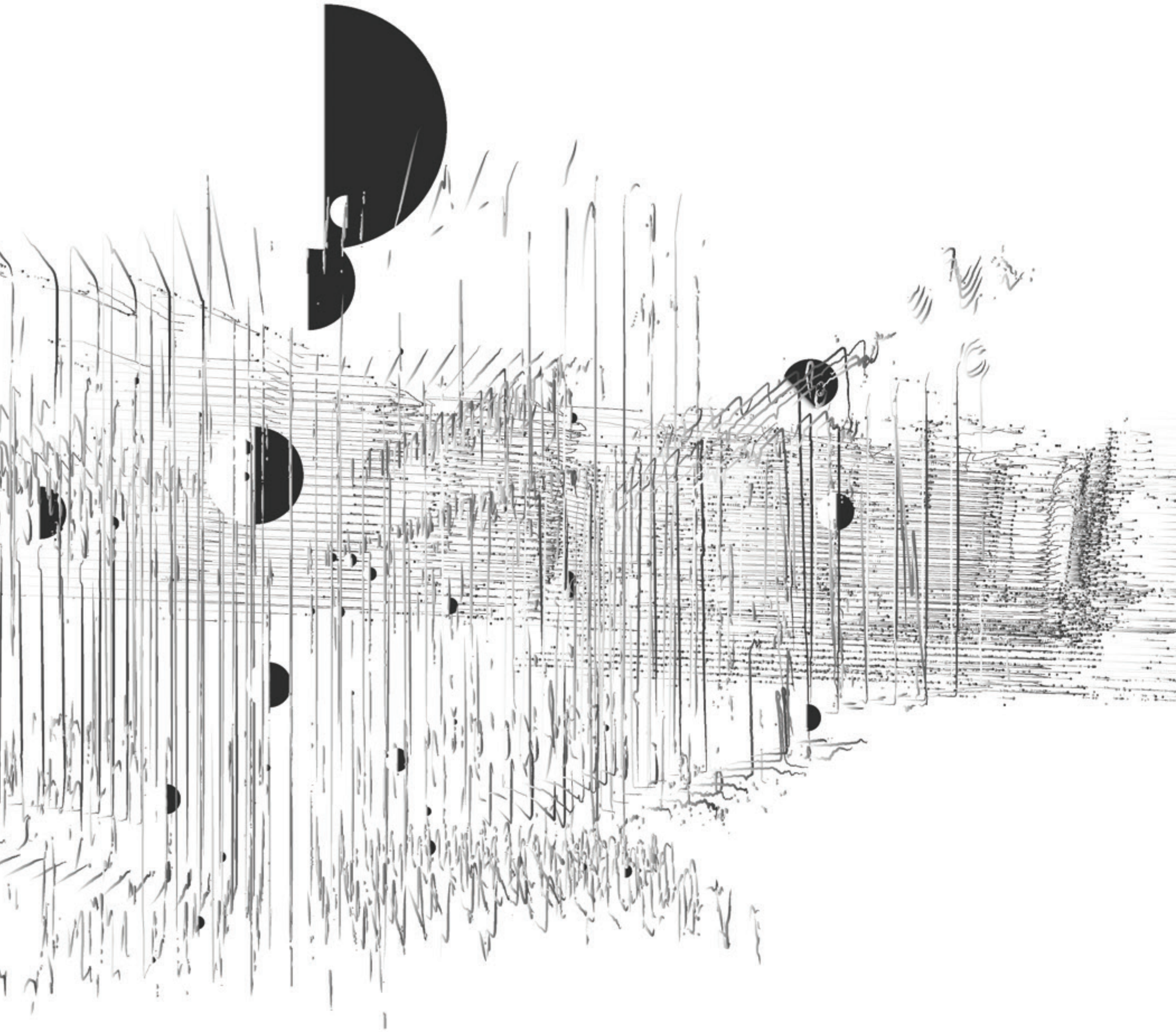
F. MYLES **SCIOTTO**

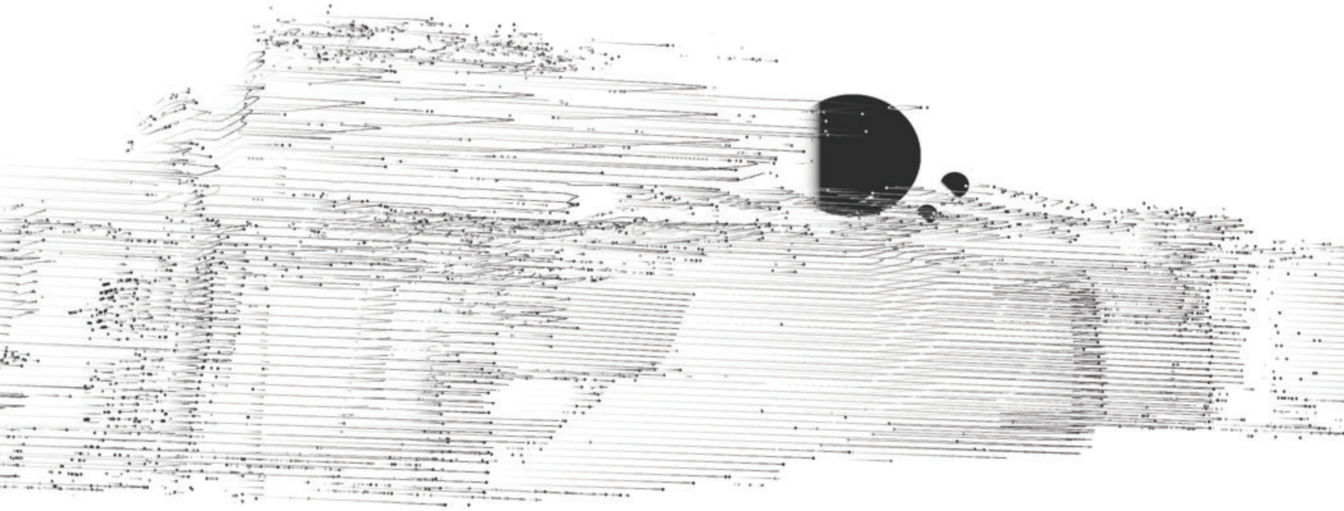
This project and accompanying proposal* experiment with the multimodal integration of sound and architecture by translating and transforming sampled sonic and spatial information in the interest of reimagining the Schindler House. The Schindler House has been approached as a place positioned delicately within nature and the balance between the composed materials and natural environment provide the source of transformative material needed to explore this trans-disciplinary concept.

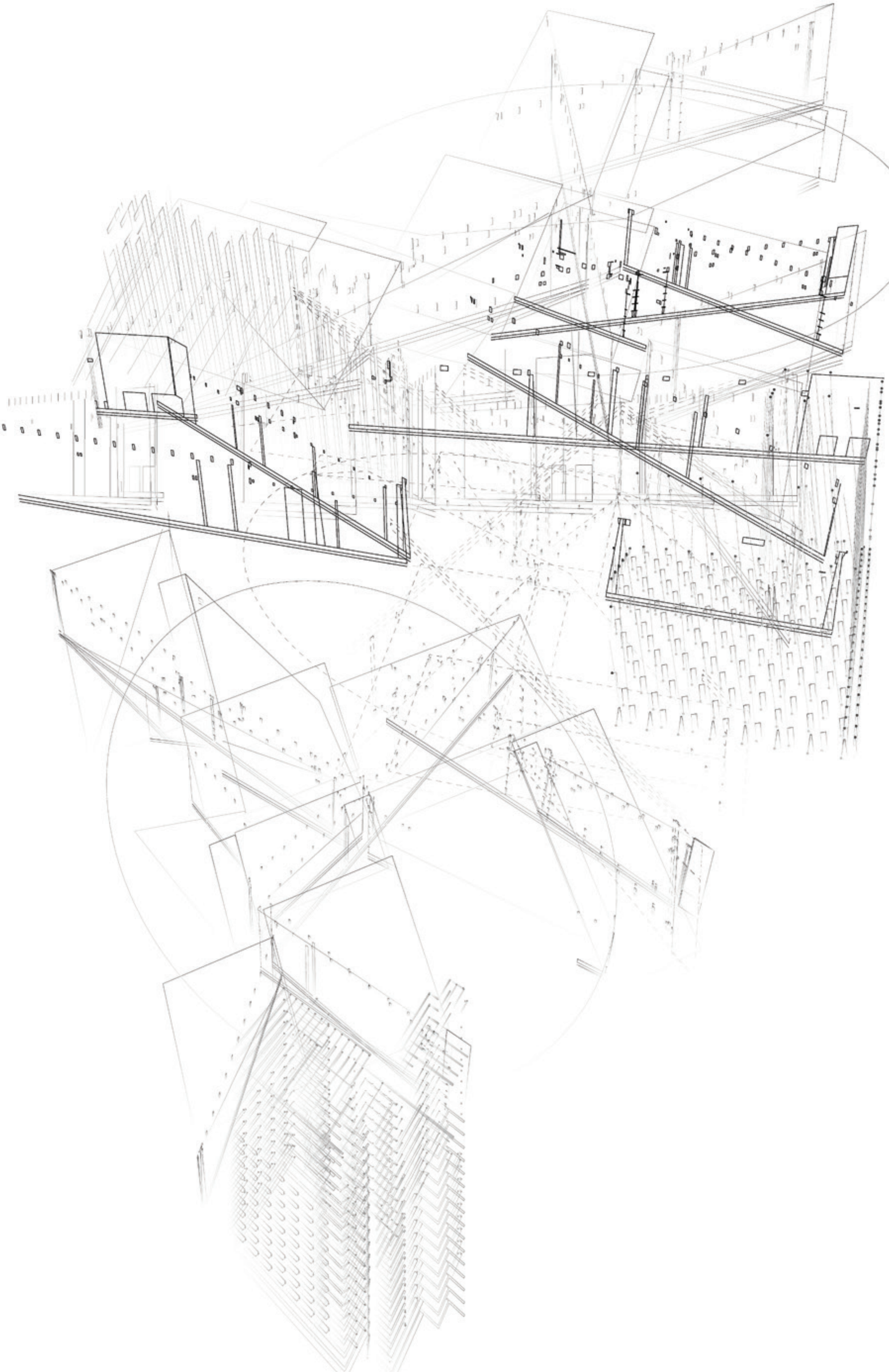
* http://schindlerlab.org/proposal/f-myles-sciotto











The methods of aural architectures and soundscapes have been introduced as related concepts for the application of sounds and sonic experience in the spatial realm. Musique concrete and its role in sampling along with transformative DSP processes have been described as techniques for the integration between the modalities of architecture and sound, generating new spatial forms throughout the unique context of the Schindler House. Questions of how sound can help us analyze and understand architectural forms related to Rudolph Schindler Kings Row House has been explored and possible directions have been outlined with the intention to continue additional in depth investigations into different processes and techniques.

Sound and the Schindler House, a spatial poem generated from a unique dialogue of proportional volumes, material fluctuations and natural contexts is scanned into patterns, sounds and rhythms, amplifying a voice inherent to this subtle and beautiful place. Schindler's "raw material of space" is transformed here into an "evolving architecture for the senses" [R.M. Schindler, Manifesto. (Vienna: 1912)].

A conversation with Curtis Roads

The EoYS team

A program like ours, at the intersection of arts and engineering, does it create a specific challenge to balance both sides?

MAT is a unique program within the University of California, Santa Barbara in that it exists at the intersection of two Colleges: Humanities and Fine Arts as well as Engineering. A main challenge of MAT is precisely to balance the arts and engineering.

In 1997, when we wrote the initial pre-proposal for MAT, we identified six areas of concentration:

1. Electronic music composition (art)
2. Sound design and music programming (engineering applied to art)
3. Visual and spatial arts (art)
4. Arts engineering (engineering applied to art)
5. Software engineering for interactive media (media engineering)
6. Media signal processing (media engineering)

Although our focus is evolving naturally, with more connection to the sciences, for example, the arts and engineering balance remains important to the MAT faculty.

The AlloSphere is the ultimate MAT rendering machine. All MAT content can be projected in the AlloSphere in virtual form.

When all these technologies being explored at MAT become part of the regular toolset of many disciplines, will programs like MAT also merge into other disciplines?

My experience with the field of computer music suggests the opposite. I became Editor of Computer Music Journal (MIT Press) in 1978 and co-founded the International Computer Music Association in 1979. I did not expect either of these to need to exist more than ten years. Yet in 2015 they are both are going strong. When I started out, the idea of combining computer science and music was considered highly interdisciplinary. Now it is a specialty. It is likely that MAT-like programs will proliferate. The concept needs to propagate down to the undergraduate level. In order for this to move forward, someone needs to write a textbook with a title like Fundamentals of Media Arts and Technology. Such a book would describe the history of the field and address the areas of concentration that I previously mentioned. This would be an excellent project for an MAT graduate to take on.

What type of future do you foresee for the so called “New” Media Arts?

MAT rides technological momentum, and as the physicist and composer Jean-Claude Risset says, we are currently in the infancy of computing. Thus MAT is very well positioned.

In MAT, we apply computation to produce powerful magic. The 17th- century English philosopher Francis Bacon defined magic as follows:

> *The science which applies the knowledge of hidden forms to the production of wonderful operations.*

In the context of the 17th century, the “knowledge of hidden forms” involved mastery of esoteric skills, analogous to the knowledge of mathematics and programming today, but also how to apply them “to the production of wonderful operations.” Criteria for the production of wonder have never been formalized. However, we observe that certain talented people make inspired choices from myriad possibilities to create fascinating designs. Computation is a means, but this remains the strong suit of human talent.

The topic of our exhibition this year is **OPEN SOURCES**. We will showcase not only the works themselves but also excerpts from their source code, to make an exhibition that is more open, more interesting, and more challenging to a technically educated public. What role does coding play in your work?

At the core of coding is interaction with technology. A modern modular synthesizer is a hybrid analog/digital computer that is programmed by means of a coding system of patch cords, buttons, switches, and potentiometers – like the original ENIAC computer of 1946.

Programming can be more widely construed as a sequence of human decisions mediated by technology, entered via text, a graphical interface, or through interactions with physical controls.

As a student I wrote generative programs for music composition based on abstract systems theory. While these were fascinating intellectually, I was not satisfied aesthetically with their output. I discovered that I was more interested in beautiful sound structures than I was in beautiful algorithms.

So I focused on programs for sound synthesis and processing, particularly granular and pulsar synthesis. I still use these in my current compositional work.

I would liken software to power tools. With power tools, you can build structures that would not be possible manually. tools as extending artistic control over domains both small that were not accessible previously.

To produce wondrous forms, what is needed is a hybrid formal/informal approach, combining the computational algorithmic control with the powerful insights of heuristics. Heuristics is the art of experience-based strategies for problem solving: "the art of good guessing."

Heuristic methods include rules of thumb, educated guesses, intuitive judgments, and common sense – all based on experience.

The heuristic approach stands in contrast to brute-force computer models that enumerate and search millions of possibilities, then make choices based on short-term statistics. Such an approach may succeed in the realm of fixed rule games like checkers and chess but have obvious limitations in the realm of art where notion of “rules” is highly context-dependent.

Heuristic methods are inevitably intertwined with an understanding of context, whether it be the state of a game, the state of a composition, or the state of a culture.

Heuristic methods are compatible with formalization. However, in practice they implement tailor-made solutions that are domain-specific and context-dependent, rather than imported whole cloth from one area of study to another. For example, the visual artist Harold Cohen has long applied heuristic algorithms to aesthetic problems. Over a period of forty years, Cohen has been developing a body of highly specific algorithms for drawing and coloring shapes.

Most importantly, heuristic algorithms are tested by experiments and refined by human perceptual judgments. One of my former teachers, the architect and composer Iannis Xenakis, used stochastic processes in a heuristic manner, sometimes modifying and rearranging the results to better suit the piece. Poetic license is the ultimate heuristic.

What is the most difficult part of keeping oneself renovated, young, and not repeating one's own work over and over?

—

I have known composers who always work a fixed number of hours per day. However, they work without inspiration most of the time, and the results are often stale and redundant. For me, downtime away from creative work is good. It is better to come to a creative space fresh. Beginner's mind is good. When I can come to a project knowing nothing that is the ideal. Then I am free to experiment and learn.

How do you decide which project to do next?

—

I always have a backlog of ideas and projects. I get new ideas all the time.

So when I finish one project there are always other interesting ones to pursue. One of the reasons I wrote my most recent book, *Composing Electronic Music* (2015, Oxford) was to think through my practice and examine what is possible and what are the important aesthetic issues now. This theoretical analysis guides my future work. I am hoping that this book will also prod open creative possibilities for readers.

You’ve said before that you never had the fear of losing inspiration, that you have worked constantly on many projects. How do you divide your time between all of them?

—

When I was writing my textbook *The Computer Music Tutorial* for MIT Press I did not compose music for fourteen years. I was not worried that my musical creativity would evaporate. Indeed, when the book was finally done I began work on my album *POINT LINE CLOUD*, which won an Ars Electronic award and led to many international performances.

To every thing there is a season. When I was Chair of MAT (2010- 2013), I essentially had to give up creative projects in order to devote full time to managing MAT. This service role turned out to be very gratifying. Working in an institution, we often find ourselves working on many matters that are urgent (deadline-driven) but not meaningful to us personally. It is only when the urgent matters are out of the way that I can focus on meaningful work. I assemble major projects over long periods of time (months and years) so I am able to chip away at them grain by grain.

The Sound Digestive System

JUAN MANUEL **ESCALANTE**

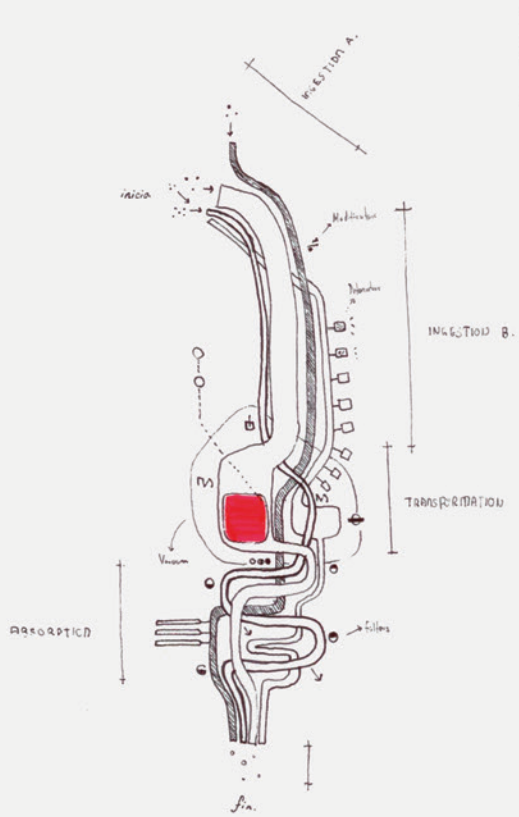
Since the second half of the 20th century, the music field opened dramatically to new sound possibilities. Specifically, the development of new technologies (from tape recorders, analog synthesizers and computers) created what we understand as electronic and digital acoustic reality.[1] During that period, the relationship of sound creation and image was also taken into account, with notable works from György Ligeti, Cornelius Cardew, Anestis Logothetis and Iannis Xenakis' UPIC System.[2]

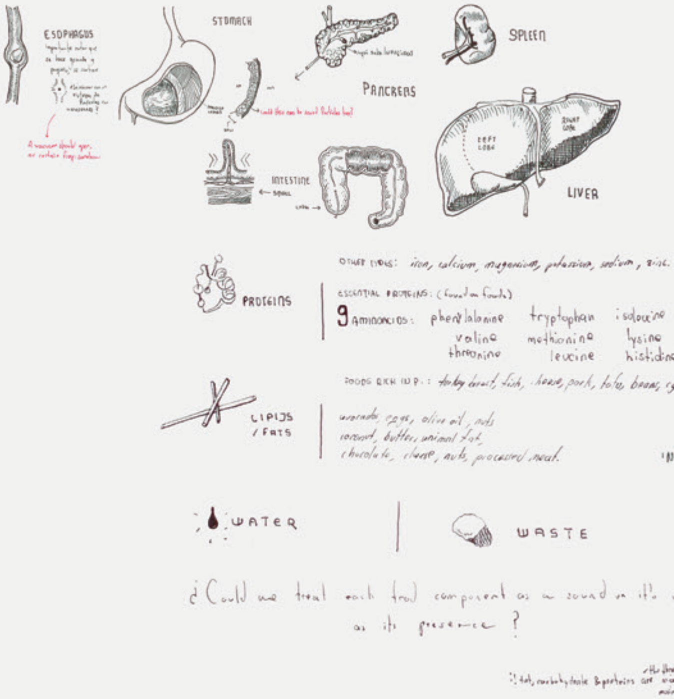
In the age of bioinformatics, bio art, and computer generated simulations in science, new ways of arranging and composing sound from these fields could open new musical and sound genres. The Sound Digestive System (SDS), proposes a new arrangement, not exclusively based on biodata, but also with a strong focus on the process and its translation into the acoustic and visual field.

[*] This project was built, designed, and composed under the guidance of electronic music composer Curtis Roads,at the Xenakis Studio at the University of California, Santa Barbara

[1] Chadabe, Joel. *Electronic Sound: The Past and Promise of Electronic Music*. Upper Saddle River, NJ: Prentice Hall, 1996.

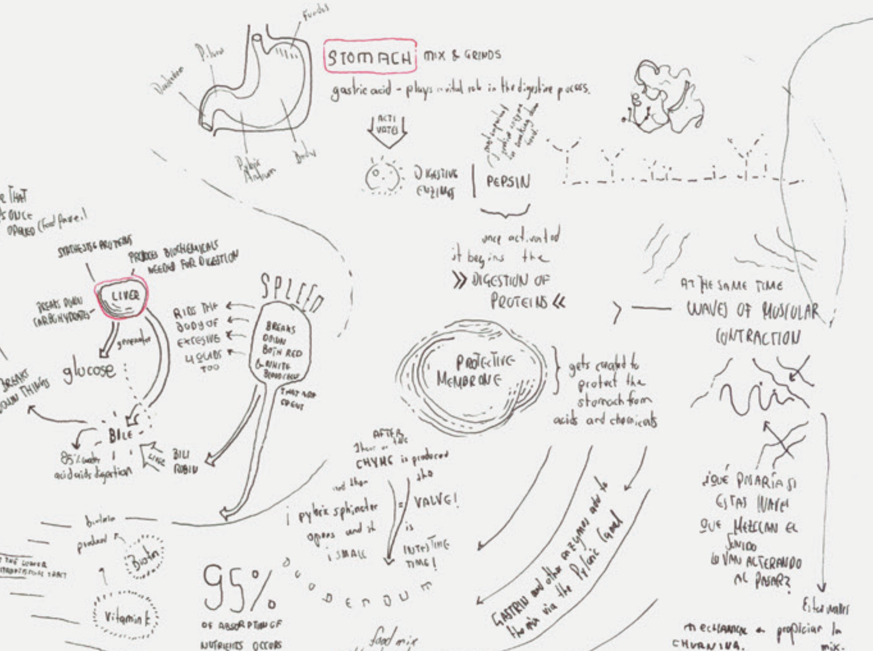
[2] Xenakis, Iannis. *Formalized Music: thought and mathematics in composition*. Prendagon Press is in Hillsdale, NY, 2014.



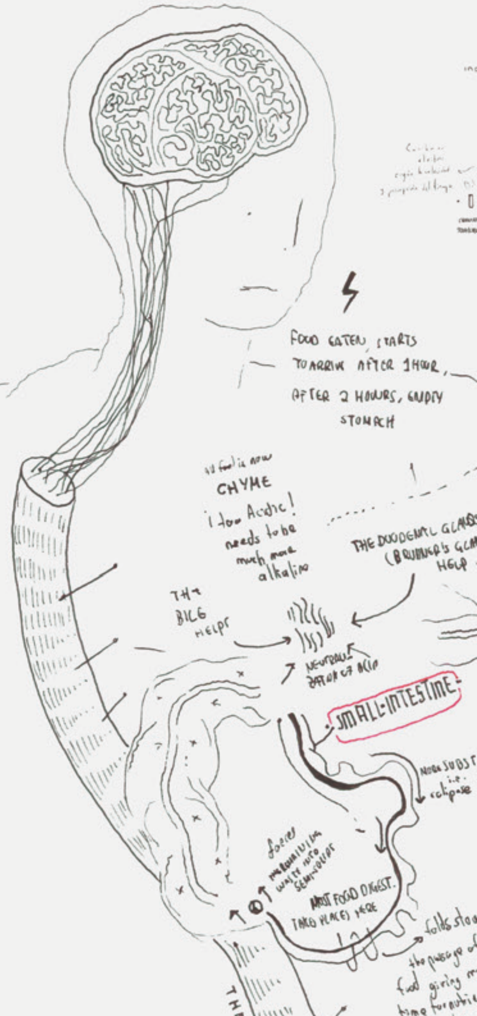


ESOPHAGUS
STOMACH
PANCREAS
SPLEEN
INTESTINE
LIVER
LEFT LOBE
RIGHT LOBE
A vacuum should open, as certain things continue
PROTEINS
OTHER ITEMS: iron, calcium, magnesium, potassium, sodium, zinc.
ESSENTIAL PROTEINS: (found on foods)
9 AMINOACIDS: phenylalanine tryptophan isoleucine valine methionine lysine threonine leucine histidine
FOODS RICH IN P.: turkey breast, fish, cheese, pork, tofu, beans,
LIPIDS / FATS
avocado, eggs, olive oil, nuts
coconut, butter, animal fat,
chocolate, cheese, nuts, processed meat.
WATER
WASTE
¿Could we treat each food component as a sound in it's
as its presence ?

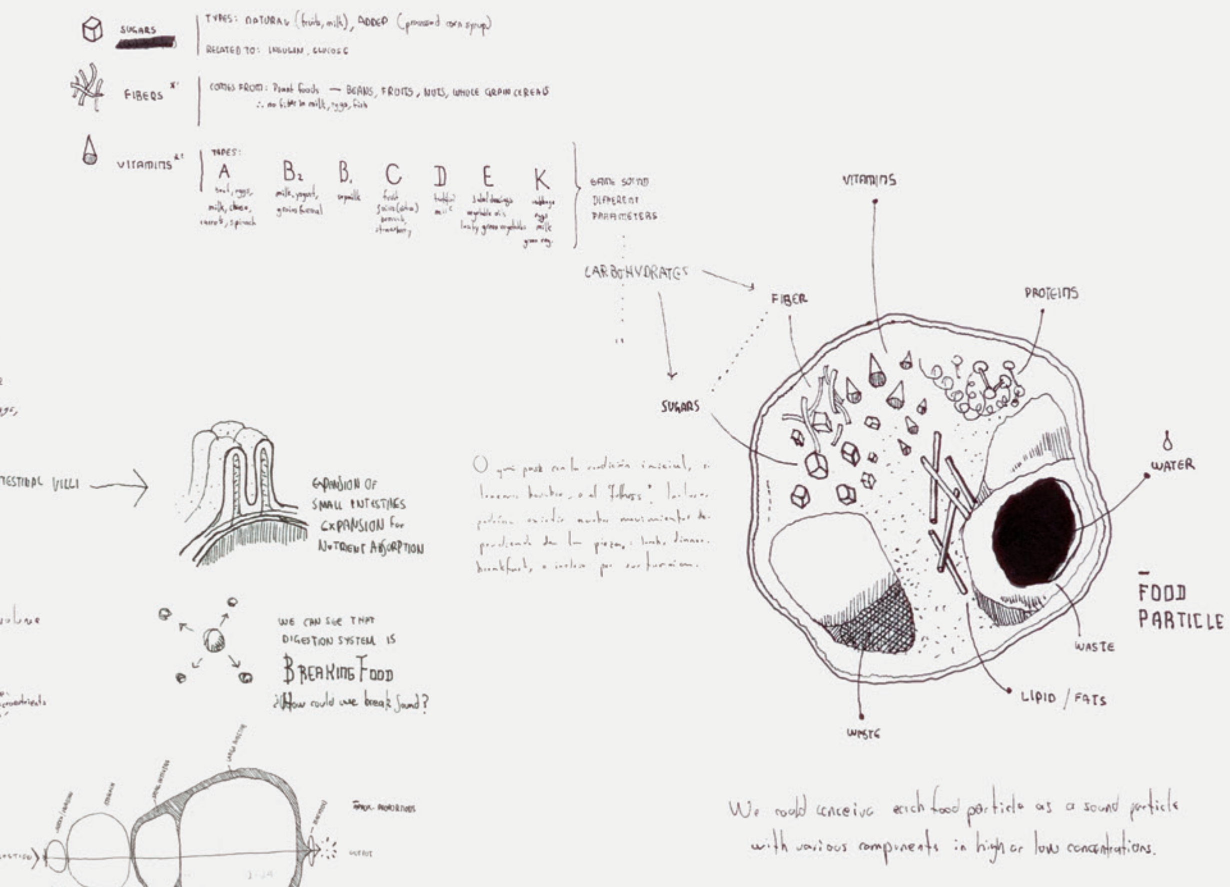
The SDS is an audiovisual piece that transforms the logic of the human digestive system into algorithmic sound composition. If we conceive this system as conduits (esophagus, gastrointestinal tract) and spaces, we can throw sound particles into these objects. A computer simulation would generate the journey of these particles as the system builds itself over time.



Fundus
Pylorus
Duodenum
Pyloric Antrum
Body
STOMACH mix & GRINDS
gastric acid - plays a vital role in the digestive process.
ACTIVATES
DIGESTIVE ENZYMES
PEPSIN
once activated it begins the
» DIGESTION OF PROTEINS «
AT THE SAME TIME WAVES OF MUSCULAR CONTRACTION
PROTECTIVE MEMBRANE
gets created to protect the stomach from acids and chemicals
LIVER
SYNTHESIZES PROTEINS
PRODUCES BIOCHEMICALS NEEDED FOR DIGESTION
BREAKS DOWN CARBOHYDRATES
generates
glucose
BILE
85% water
acid aids digestion
BILIRUBIN
SPLEEN
RIDS THE BODY OF EXCESSIVE LIQUIDS TOO
BREAKS DOWN BOTH RED & WHITE BLOOD CELLS
Biotin
Vitamin K
95% OF ABSORPTION OF NUTRIENTS OCCURS
DUODENUM
AFTER 3 hours CHYME is produced
pyloric sphincter opens and it is SMALL
VALVE!
INTESTINE TIME!
GASTRIN and other enzymes are let into the mix via the Pyloric Canal
¿QUÉ PASARÍA SI ESTAS WAVES QUE MEZCLAN EL SONIDO LO VAN ALTERANDO AL PASAR?
mechanical propiciar la mix CHURNING.



FOOD EATEN, STARTS TO ARRIVE AFTER 1HOUR,
AFTER 2 HOURS, EMPTY STOMACH
the food is now CHYME
¡too Acidic! needs to be much more alkaline
THE BILE HELPS
THE DUODENAL GLANDS (BRUNNER'S GLANDS) HELP
NEUTRALIZATION OF ACID
SMALL INTESTINE
forces remaining waste into semisolids
MOST FOOD DIGEST. TAKES PLACE HERE
folds slow the passage of food giving more time for nutrients


SUGARS
TYPES: NATURAL (fruits, milk), ADDED (processed corn syrup)
RELATED TO: INSULIN, GLUCOSE
FIBERS
COMES FROM: Plant foods — BEANS, FRUITS, NUTS, WHOLE GRAIN CEREALS
∴ no fiber in milk, eggs, fish
VITAMINS
TYPES:
A
B₂
B.
C
D
E
K
SAME SOUND
DIFFERENT
PARAMETERS
CARBOHYDRATES
SUGARS
FIBER
VITAMINS
PROTEINS
WATER
WASTE
LIPID / FATS
WASTE
FOOD
PARTICLE
INTESTINAL VILLI
EXPANSION OF
SMALL INTESTINES
EXPANSION FOR
NUTRIENT ABSORPTION
WE CAN SEE THAT
DIGESTION SYSTEM IS
BREAKING FOOD
¿How could we break Sound?
We could conceive each food particle as a sound particle
with various components in high or low concentrations.

Sound agents are divided into the following categories:

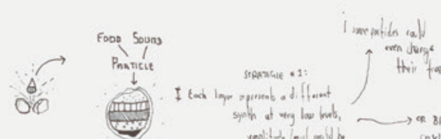
A. Sound particles (food) - A sound particle is understood as food. Each sound particle has a different composition, as each type of food presents different nutritional properties. For this study, food is divided into seven major categories:

I. Carbohydrates,[3]
II. Fibers,
III. Vitamins,[4]
IV. Proteins,
V. Lipids (not including III),
VI. Water,
VII. Waste.

Each component is assigned a different sound. At the beginning of the process, all of these sounds are mixed (uninteresting). However, as the digestion process starts, these components will be separated and treated as a separate entities. The nutritional ones will be absorbed and the waste will be disposed.

[3] Harvard School of Public Health, Carbohydrates and Blood Sugar, http://www.hsph.harvard.edu/nutritionsource/carbohydrates

[4] Harvard Health Publications: Listing of Vitamins, http://www.health.harvard.edu/newsweek/Listing_of_vitamins.htm


FOOD
SOUND
PARTICLE
STRATEGIE #1:
I Each layer represents a different
synth at very low levels,

B. Spaces (major organs) - Each organ is treated as an architectural space with reverberation properties.[5] Therefore, the scale of the stomach (experienced from the inside) would be radically different than, for example, a liver or a spleen.

C. Conduits (connections between spaces) - Transitional spaces between major organs have the same importance as transitional passages in music composition. From the biological point of view, important stages of the process occur at these steps. For example, pancreatic enzymes along with substances from the gallbladder join the digestive mix in the duodenum. For this piece, the esophagus is also treated as a conduit.

D. Processes / Actions - Food breaks and changes its own composition, excessive water is handled by the spleen, an overdose of sugar causes chemical reactions in the liver and pancreas and so forth. All of these operations are transformed into sound operations: substraction, multiplication, addition, pitch and phase shifting.

:: Sound Synthesis

This project is conceived as a live-perfomance piece where sound and image are being processed in real time. This means, according to the type of food thrown into the system during a specific performance, it generates a different experience every time.

For sound synthesis, *SuperCollider,* an environment and programming language by James McCartney[6] is used. Each synthesizer (or *synth*) can produce one or three sounds. Depending on the code of each *synth*, different sound generators (also known as unit generators: UGens), envelopes, and other types of parameters could be assigned. The total number of *synths* for this piece is 21.

Each *synth* – or variations of a given one – was assigned to the following objects: esophagus walls, breaking food, stomach enzymes, stomach contractions, blue glucose, breaking white/red cells, pancreatic juice, basal electrical rhythm, rectum to brain signal, release, upper gastro intestinal tract, lower gastro intestinal tract and each component of a food particle (carbohydrates, fibers, vitamins, proteins, fats, water and waste).

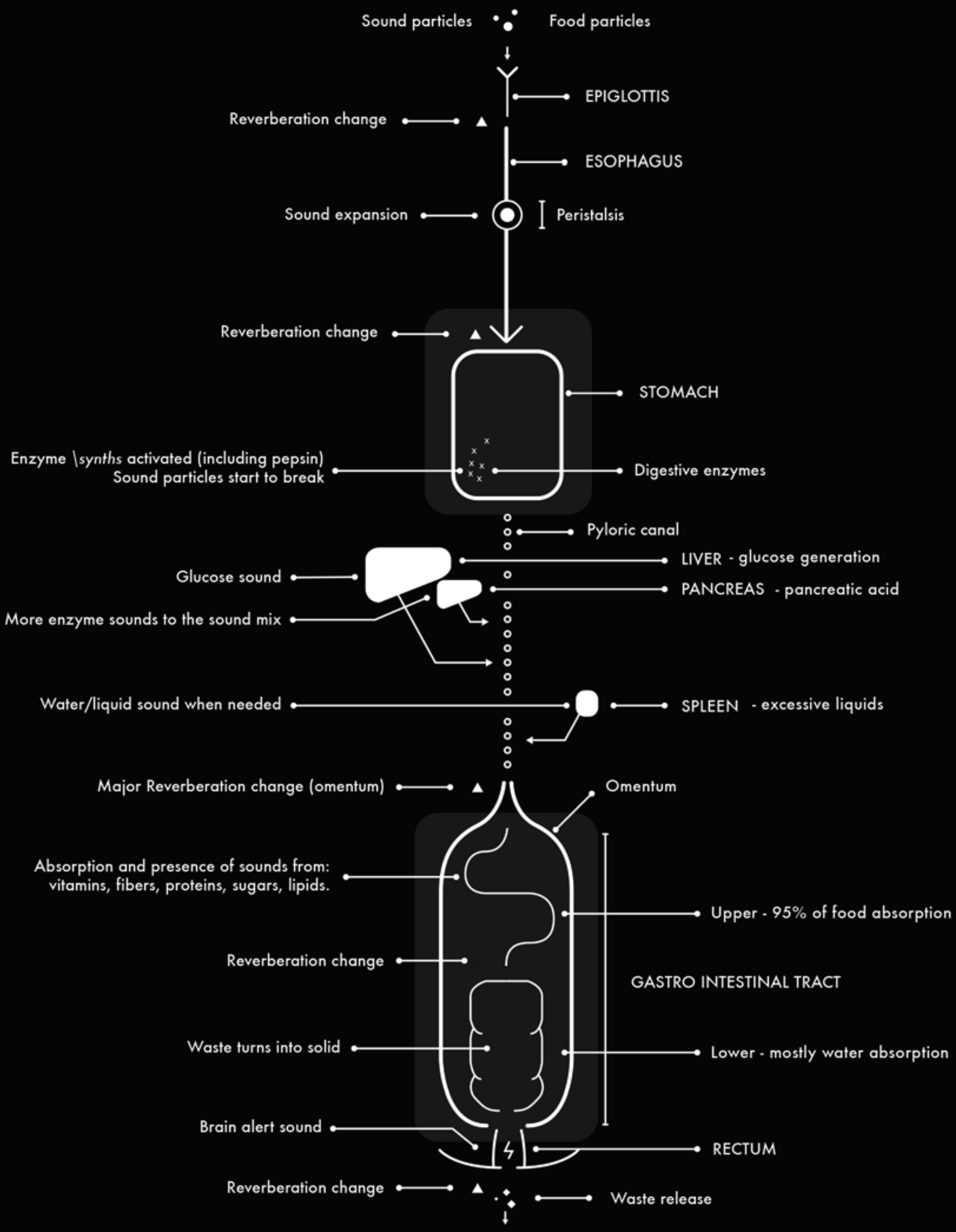
:: Graphic Score

The arrangement of sounds over time is controlled with the open source software Processing. Three main libraries are used for the project: *SuperCollider*, *oscP5*[7] and *netP5* Visual cues distributed in space (represented in two dimensions) would trigger a specific reverberation, or sound particle transformation, according to its own position (which is also a visual approximation of the whole system). Visual cues positioned higher vertically would represent an early stage of the sound digestive process, while sounds positioned lower vertically are closer to full absorption and waste release.

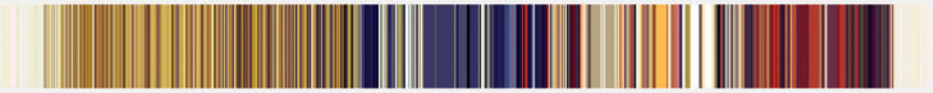
[5] Roads, Curtis. *The Complete Music Tutorial*. Cambridge, MA, MIT Press, 1996.

[6] SuperCollider library for Processing developed by Daniel Jones, http://www.erase.net/projects/processing-sc/

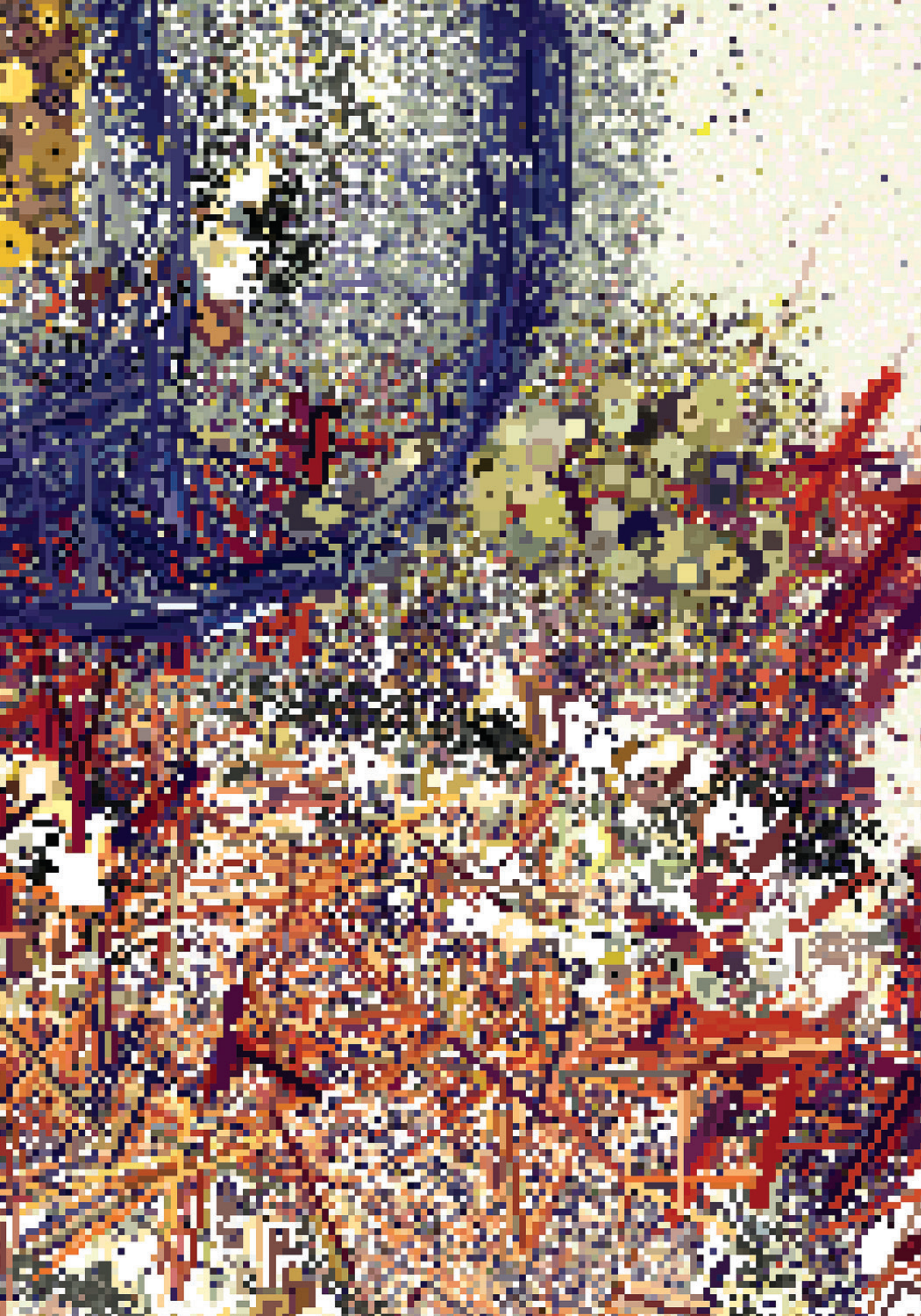
[7] *OscP5* library for Processing written by Andreas Schlegel, based on Open Sound Control (OSC) protocol by Wright, M. and Freed, A. http://www.sojamo.de/libraries/oscP5/


Sound particles
Food particles
EPIGLOTTIS
Reverberation change
ESOPHAGUS
Sound expansion
Peristalsis
Reverberation change
STOMACH
Enzyme *synths* activated (including pepsin)
Sound particles start to break
Digestive enzymes
Pyloric canal
LIVER - glucose generation
Glucose sound
PANCREAS - pancreatic acid
More enzyme sounds to the sound mix
Water/liquid sound when needed
SPLEEN - excessive liquids
Major Reverberation change (omentum)
Omentum
Absorption and presence of sounds from:
vitamins, fibers, proteins, sugars, lipids.
Upper - 95% of food absorption
Reverberation change
GASTRO INTESTINAL TRACT
Waste turns into solid
Lower - mostly water absorption
Brain alert sound
RECTUM
Reverberation change
Waste release

```
class Organ {

  // Properties %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  // Switches
  float velMin = 0.03;
  float velMax = 1.4;
  //Vars:
  int cantidad, progressions;
  float scale, halo, friccion, desplazamiento;
  float XF, YF; // Origin coordinates
  int [][] pointsO;// Originales
  int [][] pointsF;// Finales (deseo de la fricción)
  int [][] points;// Los que se colocan
  Boolean [] onData; // Gradual
  color [] pointsColors; //Colores
  float [] angulos;
  float [] sentidos;
  float [] angulosB;
  float [] vel;
  int [] reactionCounter;
  float organSize;
  PImage organLoc, imgColor;


  // Constructors and Helpers %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   Organ(PImage organLoc, PImage imgColor, int cantidad, float scale, float halo, float friccion, float desplazamiento,
float altura, int progressions) {  }
  int [] getPosition( int position, PImage [] progressions ) {  }
  int getProgressions() {  }
  PImage getOrganLoc() {  }
  int [] getPosicionAzar() { }
  void setInteraction( float x1, float y1 ) {   }
  void calculateFriction( ) {  }


  //  Organ Methods %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  int [] getPuntoNegro(PImage imagen) {  }
  void helperImage() {  }
  void showStomach() {  }
  void showUpperGastro() {  }
  void showLowerGastro()  {  }
  void showLiver()  {  }
  void showPancreas()  {  }
  void showSpleen()  {  }
  void showRectum()  {  }

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

}
```



A brighter version of *Annunciazione* by Leonardo da Vinci was used as a color database. Leonardo's notable sense of proportion and balance, is used – in this case – to distinguish each organ and at the same time, achieve equilibrium in the whole composition. Each organ is programmed to grab colors from a specific area/object of the painting.



Localization on Google glass

CHRIS **SWEENEY** · BENJAMIN **NUERNBERGER**

We present a new algorithm for using knowledge of the vertical direction (or alternatively, the gravity direction) to compute the absolute pose of a camera. Our method is general and works for single or multi-camera systems. Algorithms that utilize inertial sensors that are increasingly important as these sensors become more common on imaging devices. Utilizing motion priors reduces the number of unknown degrees of freedom to solve for and reduces computational complexity. This is important for augmented reality applications where computational resources are often limited.

Our algorithm utilizes an angle-axis formulation which has been proven previously to be very robust to noise. Indeed, our algorithm has slightly better robustness to image and IMU noise compare to the current state-of-the-art. We also show that our algorithm gives better accuracy in a localization experiment with an iPhone 4. Perhaps most importantly, our algorithm only involves solving a quadratic equation and is 3 orders of magnitude faster than the current state-of-the-art. We show that this increased efficiency allows our algorithm to be used for localizing frames in a video sequence directly from a reference 3D model in real time. Finally, we created an augmented reality application using a Google Tango tablet to show the real-world feasibility of our algorithm. We are eager to use our new method on other devices, and have already begun development of our algorithm for a Google Glass device.

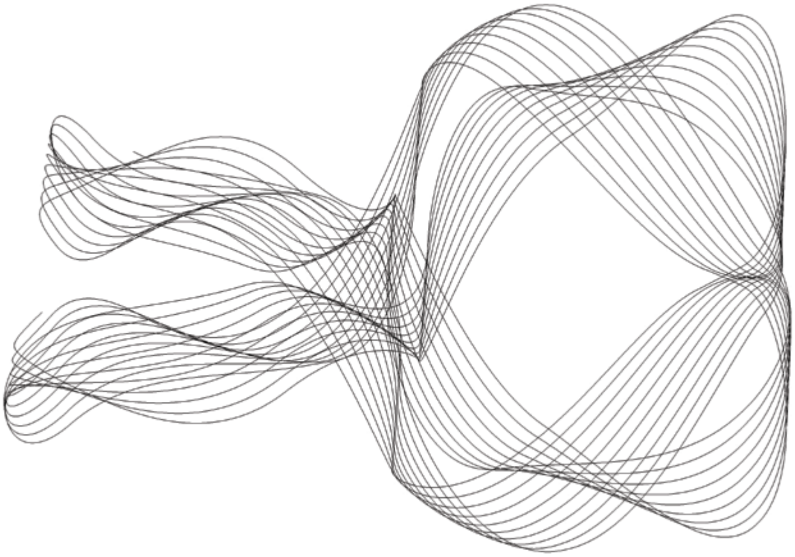
Kinetic V2

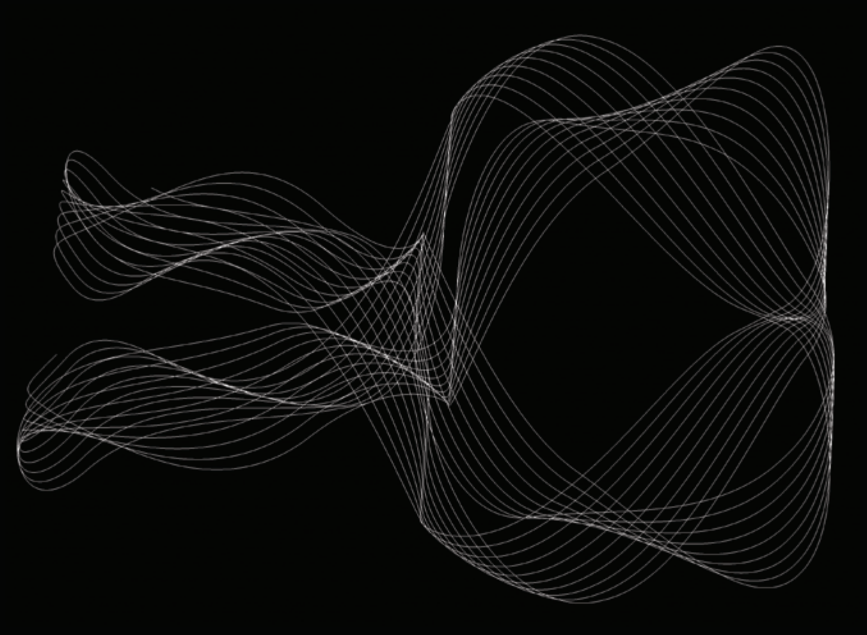
RYAN **MCGEE**

Kinetic V2 is the second work composed using Spatial Modulation Synthesis, a novel AM/FM/Granulation technique using simulated high-speed periodic sound trajectories to produce and morph timbres. The graphics are not a visualization of the sounds, but rather a tracing of their spatial orbits as Doppler shift, gain attenuation, and multi-channel spatialization warp their sonic characteristics. Kinetic V2 fully utilizes the full-surround 3D graphics of the AlloSphere along with the entire 54.1 channel sound system to completely immerse the listener in a unified audio-visual experience.

ryanmcgee.com

```
void SoundSource::move()
{
    motionOsc.x += direction.x*modIndex.x/M_PI;
    motionOsc.y += direction.y*modIndex.y/M_PI;
    motionOsc.z += direction.z*modIndex.z/M_PI;

    double thetaX = motionOsc.x < 0 ? (motionOsc.x/lowerBounds.x)*2*M_PI : (motionOsc.x/
upperBounds.x)*2*M_PI;
    double thetaY = motionOsc.y < 0 ? (motionOsc.y/lowerBounds.y)*2*M_PI : (motionOsc.y/upperBounds.y)*2*M_
PI;
    double thetaZ = motionOsc.z < 0 ? (motionOsc.z/lowerBounds.z)*2*M_PI : (motionOsc.z/upperBounds.z)*2*M_
PI;

    double xVel = abs(sin(thetaX));
    double yVel = abs(sin(thetaY));
    double zVel = abs(sin(thetaZ));

    xVel *= modIndex.x * direction.x;
    yVel *= modIndex.y * direction.y;
    zVel *= modIndex.z * direction.z;

    velocity.set(xVel, yVel, zVel);

    Vec3d newPos = pos() + velocity;

    pos(newPos.x, newPos.y, newPos.z);

    //history
    trail.write(pos());

    //bounds
    if(POS::pos().x >= upperBounds.x) direction.x = -1;
    else if(POS::pos().x <= -lowerBounds.x) direction.x = 1;

    if(POS::pos().y > upperBounds.y) direction.y = -1;
    else if(POS::pos().y < -lowerBounds.y) direction.y = 1;

    if(POS::pos().z > upperBounds.z) direction.z = 1;
    else if(POS::pos().z < -lowerBounds.z) direction.z = -1;
}
```





Attention-Aware User Interfaces

MATHIEU **RODRIGUE**

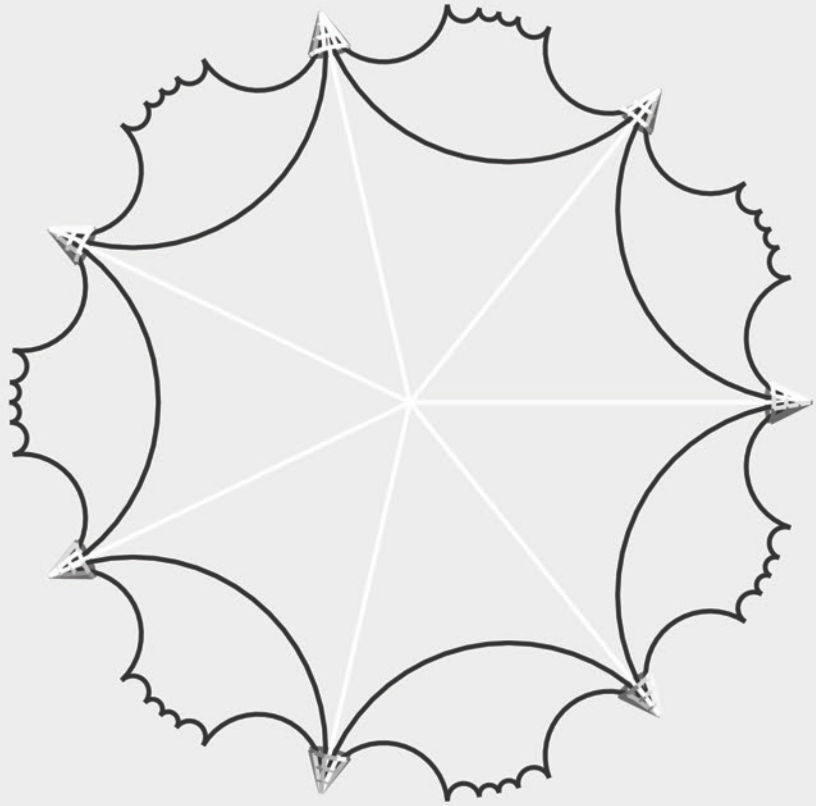
Attention-aware user interfaces detect attention states in users during learning tasks. Eye tracking and electroencephalography (EEG) monitor the user during reading and provide a classifier with data to decide the user's attention state. The multimodal data informs the system where the user was distracted spatially in the user interface and when the user was distracted in time, allowing the option for future real-time systems to facilitate learning when appropriate.

```
EyeTracker::EyeTracker()
{
   // Connect to the server in push mode on the default TCP port (6555)
   if( m_api.connect( true ) )
   {
       // Enable GazeData notifications
       m_api.add_listener( *this );
   }
}
```

```
EyeTracker::~EyeTracker()
{
   m_api.remove_listener( *this );
   m_api.disconnect();
   fclose (gazeLog);
}
```

```
void EyeTracker::openLogFile(std::string logFile)
{
std::string log = “eyeLog.txt”;
fopen_s(&gazeLog, log.c_str(), “w”);
}
```

```
void EyeTracker::on_gaze_data( gtl::GazeData const & gaze_data )
{
pupilSize = (gaze_data.lefteye.psize + gaze_data.righteye.psize)/2;
eyeX = gaze_data.raw.x;
eyeY = gaze_data.raw.y;
eyeBlink = gaze_data.state & gaze_data.GD_STATE_TRACKING_GAZE;
}
```



Versor

PABLO **COLAPINTO**

Versor is a library of code for spatial computing, and is based on a rigorous mathematical system known as geometric algebra. Developed with the intention of helping artists, engineers and scientists collaborate, Versor accentuates visual and constructive thinking.

The mathematics of geometric algebra enable users to formulate relationships between geometric primitives like circles and spheres. For instance, the circle interesection of two spheres can be expressed as:

$$\kappa = \sigma_1 \wedge \sigma_2$$

where σ_1 and σ_2 are spheres. We call this intersection of two spheres a coincidence relation, since the spheres are coincident on the circle. The similarity of the "exterior" operator $\wedge$ with the logical operator "and" is intentional, and reveals the logical nature of geometric algebra: it combines deductive and inductive reasoning into one spatial computing engine.

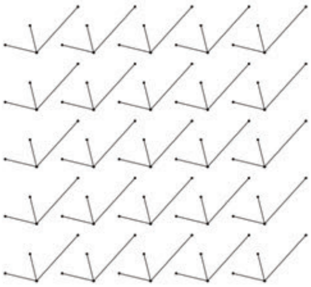
The foundations of this approach to navigating the topology of forms lie in Hermann Grassman's theory of extension first published in 1844. William Clifford later developed Grassman's exterior algebra into a geometric algebra by lifting into onto a metric, thereby generalizing William Hamilton's quaternions. Clifford's particularly illuminating way of calculating with spatial forms was left to relative obscurity until the 1960's, when the physicist David Hestenes published a series of texts exploring isomorphisms of Clifford's algebra with the prevalent models of quantum mechanics. Exploring alternative metric spaces, Hestenes showed that the algebra could be used to model all geometric transformations, such as rotations, translations, twists, dilations, and bends. For instance, the rigid twisting motion around a line λ can be expressed as an exponential:

$$\mathcal{M} = e^{\lambda}$$

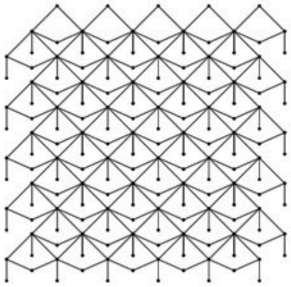
Once this expressivity of motion was articulated, geometric algebra blossomed into fields as diverse as robotics, microbiology, optics, astronomy, and computer graphics, attracting researchers interested in developing intelligent systems. Leo Dorst, Eduardo Bayro-Corrochano, and Anthony and Joan Lasenby are some of the leading researchers in the still somewhat specialized field of geometric algebra.

Implementations of Versor exist in C++, Lua, and JavaScript, with Python forthcoming. In C++, the core of the library is only 150kb of code: using template metaprogramming, the work of figuring out what geometric primitive types are created by more complex expressions is left to the compiler, which enables a way to use the computer to both generate geometry and identify potential algorithms with which to construct relationships. While the library's core utilities focus on the conformal model of 3D Euclidean space, it can generate algebras in arbitrary dimensions and arbitrary metrics. In the pages that follow, we examine some complex spatial transformations that are greatly simplified with this system.

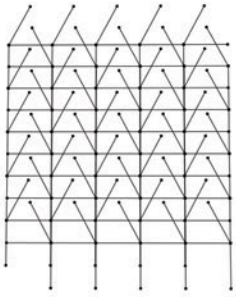
The word "versor" itself is an element of geometric algebra, and represents a reflection, the simplest transformation from which all others descend.



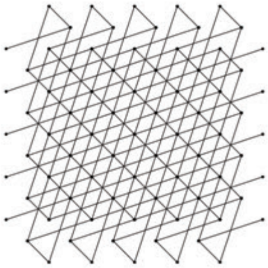
OBLIQUE p1



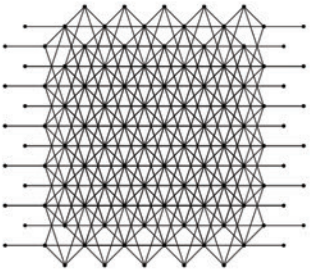
RECTANGULAR cm



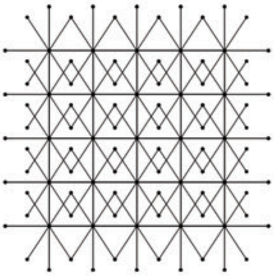
RECTANGULAR pg



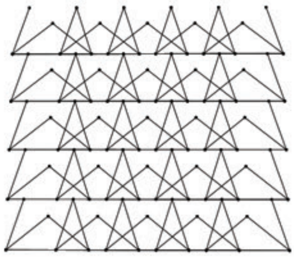
OBLIQUE p2



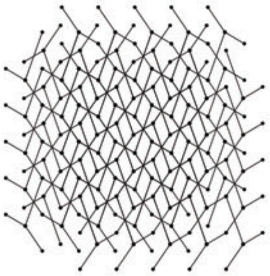
RECTANGULAR cmm



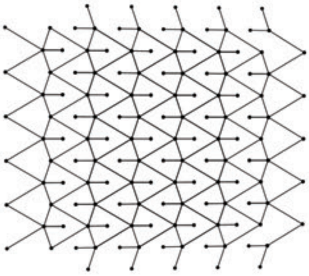
RECTANGULAR pmm



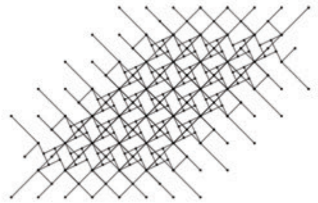
RECTANGULAR pm



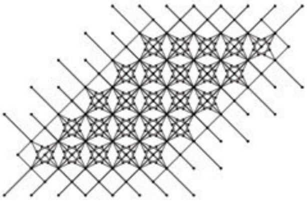
RECTANGULAR pgg



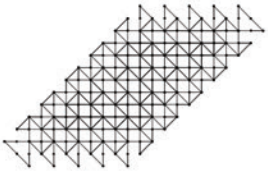
RECTANGULAR pmg



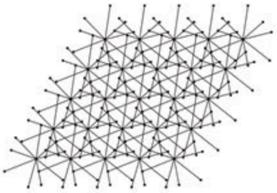
SQUARE p4



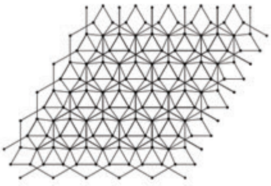
SQUARE p4m



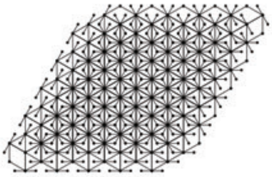
SQUARE p4g



TRIGONAL p3

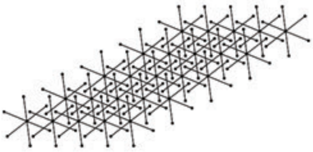


TRIGONAL p3m1

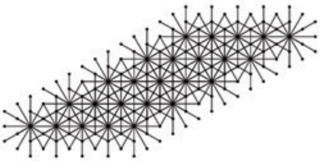


TRIGONAL p31m

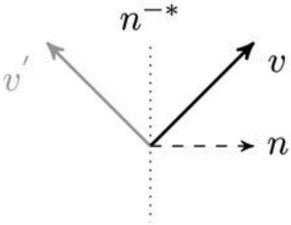
2D crystallographic space groups
(also known as the Wallpaper groups).



HEXAGONAL p6

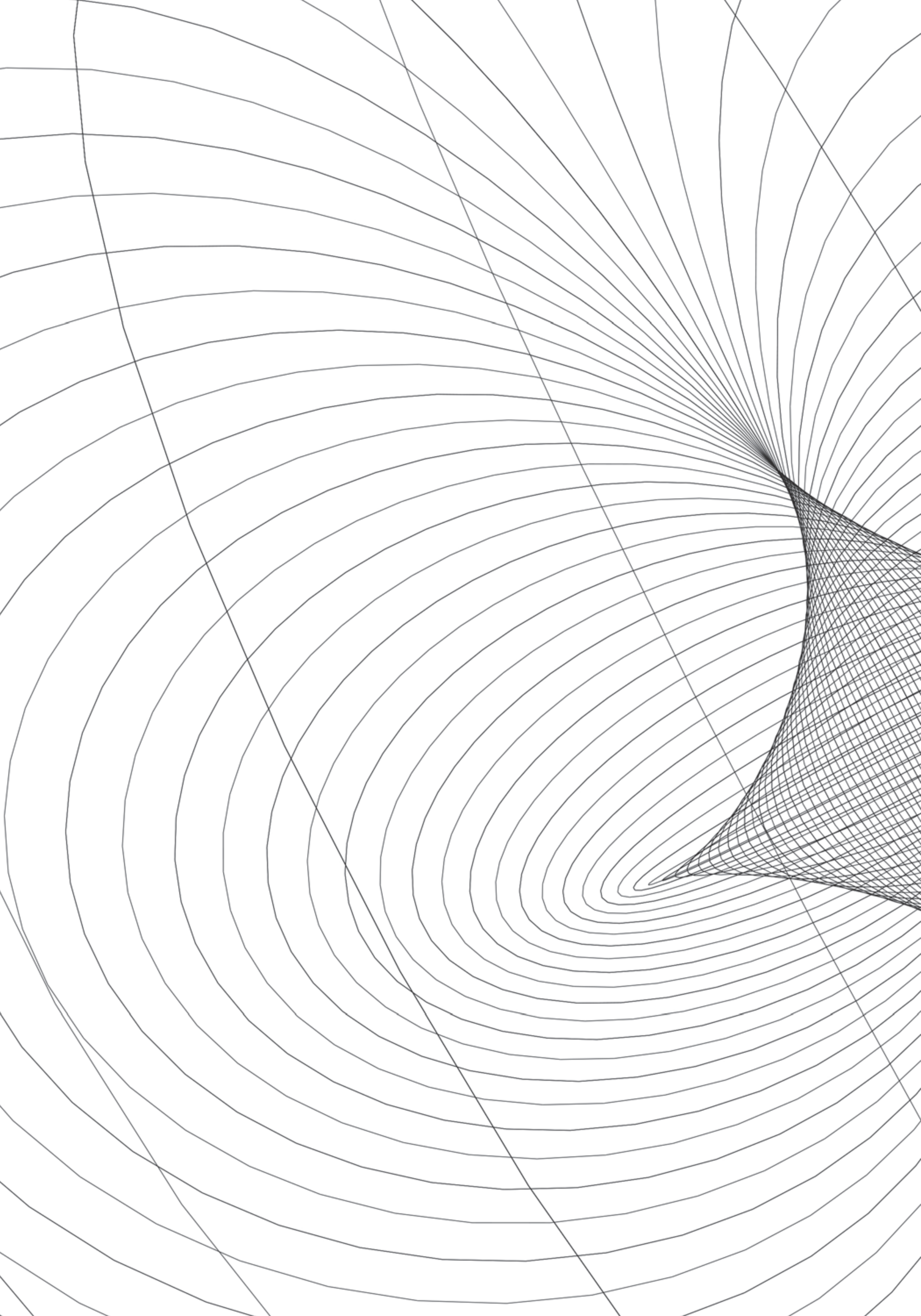


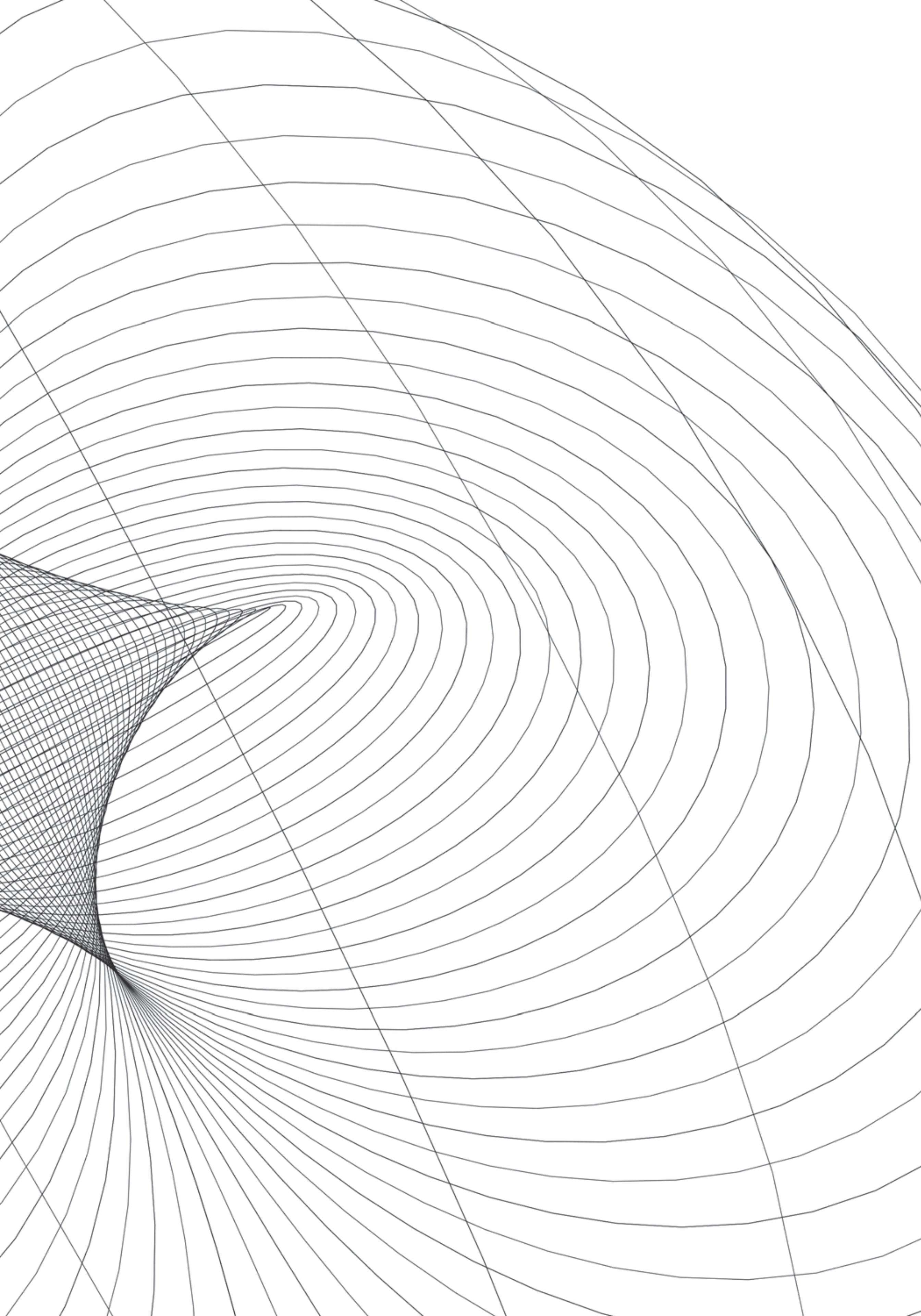
HEXAGONAL p6m



REFLECTION: v --> v^1 = -nvn

As shown by D. Hestenes and J. Holt in *The Crystallographic Space Groups in Geometric Algebra* (2007), the conformal model provides a natural foundation for generating symmetry groups from versor reflections. Rotational symmetries such as *p2* and *p3* are generated by the composition of two reflections. Glide reflections such as in *pgg* and *pmg* are composed of a reflection and a translation.





kurtkaminski.com

younkhg.wordpress.com

Vena

KURT **KAMINSKI** · KEEHONG **YOUN**

Much of life on Earth depends on the success of plants and their relationship with soil. Obscured from sight, delicate and complex patterns emerge as the plant searches for nutrients. Vena is a study of the architecture of this nutrient gathering. Based on Kurt Kaminski's Allosystem implementation of a three dimensional leaf venation algorithm, Keehong Youn joins as collaborator and co-developer on the expanded version. Vena focuses on fully utilizing the Allosphere's stereoscopic omniprojection as tools for experiencing growth as space. By abstracting soil nutrients as resources and plant roots as consumers, a broader tension arises as growth becomes both overwhelmingly complex and limited by attainable material. The spatial realization is at once beautiful and disturbing, revealing the claustrophobic scaffolding which underpins life's scramble for existence.

Described by Przemysław Prusinkiewicz in his 2007 paper, “Modeling Trees with a Space Colonization Algorithm,” the algorithm works in roughly four steps. First, branch segments are placed as a root or axiom. Then, attractor points are scattered, each point aware of the closest branch within a given radius. If an attractor is close enough, new segments are created from the previous segment. Their growth direction, and ultimately the overall form, is determined primarily by the length of the individual segments and the minimum proximity to an attractor before that resource is considered exhausted. The process repeats until all the nutrients are depleted or too far away.

```
////////////////////////////////////////////////////////////////////////////////
// process the leaves : find closest branch, add dir to leaf
////////////////////////////////////////////////////////////////////////////////
for (int i = 0; i < leaves.size(); i++) {
  Leaf& li = leaves[i];
  if (li.skip) continue;

  li.ClosestBranch = NULL;
  float curr_min_dist = maxDistance; // start with max dist, save anything closer
  Vec3f min_dir = Vec3f(0, 0, 0);

  // find nearest branch for this leaf
  for (int j = 0; j < branchVec.size(); j++) {
    Branch* b = branchVec[j];
    Vec3f direction = li.Position - b->Position;
    float distance = direction.mag();

    // skip this leaf next time if branches are too close. no more leaf growing!!
    if (distance <= minDistance) {
      li.skip = true;
      break;
    }

    // branch is in range, determine if it’s the closest
    if (distance <= curr_min_dist) {
      li.ClosestBranch = b;
      curr_min_dist = distance;
      min_dir = direction;
    }
  }

  // tell the branch: this leaf would like it to grow, and in what direction
  if (!li.skip && li.ClosestBranch != NULL) {
    Vec3f dirNorm = min_dir.normalize();
    li.ClosestBranch->GrowDir += dirNorm;
    li.ClosestBranch->GrowCount += 1;
  }
}
```

```
////////////////////////////////////////////////////////////////
// generate new branches
////////////////////////////////////////////////////////////////
newBranchesVec.clear();

for (int i = 0; i < branchVec.size(); i++) {
  Branch* b = branchVec[i];

  // since we are looping through all the branches now, it's a good time to increment color
  Color plus_width = RGB(b->Width, 0, 0);
  m_tree.colors()[i*2] += plus_width;
  m_tree.colors()[(i*2)+1] += plus_width;

  if (b->Skip) continue;

  // if at least one leaf is affecting the branch
  if (b->GrowCount > 0) {
    Vec3f avgDirection = b->GrowDir / b->GrowCount;
    avgDirection.normalize();

    // set grow count to 0 so the new branches don't inherit a grow count > 0
    b->Reset();

    // create a branch with the new position info
    Branch* newBranch = new Branch(b, b->Position + avgDirection * branchLength,
avgDirection);
    newBranchesVec.push_back(newBranch);
  }
}
```

```
////////////////////////////////////////////////////////////////////////
// add new branches to tree
////////////////////////////////////////////////////////////////////////
bool branchAdded = false;

for (int i = 0; i < newBranchesVec.size(); i++) {
  Branch* b = newBranchesVec[i];

  m_tree.vertex(b->Parent->Position);
  m_tree.vertex(b->Parent->Position);

  m_tree.color(treeInitialColor);
  m_tree.color(treeInitialColor);

  newPos_tree.push_back(b->Position);

  b->Width = 0.0001;
  b->group = growthIteration;
  b->siblings = newBranchesVec.size();

  branchVec.push_back(b);
  branchAdded = true;

  Branch* p = b->Parent;
  while (p->group >= 0) {
    p->Width += .0001;
    p = p->Parent;
  }

}
```

```
public class UseRenderingPlugin : MonoBehaviour {

    int width = 1024;

    int height = 576;

    ...


    private IEnumerator CallPluginAtEndOfFrames ()

    {


        while (true) {

            if(Input.GetKeyDown("f"))
            {
                Screen.fullScreen = !Screen.fullScreen;

            }

            if(Input.GetKeyDown ("q"))
            {

                endServer ();
                Application.Quit ();

            }


            // Wait until all frame rendering is done
            yield return new WaitForEndOfFrame();


            //Need to call this (or any plugin function) to keep calling native rendering events
            SetTextureFromUnity (tex.GetNativeTexturePtr());


            //movingCamera.Render();
             RenderTexture.active = movingCamera.targetTexture;

            tex.ReadPixels(new Rect(0, 224, width, height), 0, 0);


            //tex.ReadPixels(new Rect(0, 0, width, height), 0, 0);
            RenderTexture.active = null;
            tex.Apply ();

            GL.IssuePluginEvent (1);

        }


    }


}
```

Mixed Reality Simulation

TIBOR **GOLDSCHWENT** · MATHIEU **RODRIGUE**
DREW **WARANIS** · TIM **WOOD** · TOBIAS **HÖLLERER**

By means of a large (surround) virtual reality display, augmented reality (AR) applications and devices are simulated. By doing so, user studies – evaluating the interaction methods of the AR applications – as well as the exploration of fundamental AR characteristics will be facilitated. Furthermore, even the simulation, and thus the investigation, of innovative, currently unrealizable AR applications will be enabled.

A conversation with

JoAnn Kuchera-Morin

■ ▲

Fabian **Offert** • Juan Manuel **Escalante**

Transcription by Kurt Kaminski

Could you talk a bit about the challenge of running a program that is specifically designed to fuse art, engineering, and science? What are the challenges that persist, what are the challenges that we have overcome, and what is your vision for those challenges?

That is a great question because that has been thirty years of my life right now, starting as a composer of orchestral music and getting my degree from a conservatory – the Eastman school of music – and having a *PDP-11* computer dropped into our lab in 1981. I was being trained and educated with traditional tools, and there were things that I wanted to do with the traditional tools that I could not do. All of a sudden this thing, like a beast or monolith, was put in front of me and gave me the opportunity to actually transform, for instance, my concert hall into the inside of a Stradivarius.

This, however, put me into a position that was at the same time interesting and horrible. When I came here in 1984, as a young assistant professor, I realized that I had received a degree – a wonderful degree – that had not educated me for the future. Being thrown in the college of engineering and being told, as a composer, that I had to build the first digital signal processing system together with my colleagues in electrical engineering and computer science, was, indeed, a big challenge. When I walked into the door of Harold Frank Hall, they gave me a closet on the second floor. I will never forget the head of the computing center coming up to me and saying:

> "What is this thing? This piece of hardware?"
> And I said, "You mean the digital-to-analog converter?"
> "Yes, what does it do."
> I said, "It turns numbers into sound so that you can hear your data."
> And he goes, "Why would anybody want a computer to make sound."

But then we started talking about the challenges of bandwidth, and I envisioned a day that people would want to use computers for visuals. I said, that we wanted to do signal processing in real-time, with hundreds of channels, with instrumentalists on the east coast and the west coast playing together.

Mind you, this was before the Internet. They asked me if I was insane, and I said yes, I am, but functional and: content needs to drive technology.

Speaking of content driving technology, what is the role of a program like MAT within the general struggle between art and industry: Jeffrey Shaw inventing virtual reality, and now we have the Oculus. How can a program like MAT add to the persistence of the arts against this type of take-over?

This is exactly what we're all about. The reason why we made the move in the 2000s was to work with our materials scientists and to come into the Nanosystems Institute. When you start to talk about the challenges and partnerships going through these interdisciplinary relationships and you start to understand the idea of form and function and instrument design and building, then you start to understand the relevance of who we are, and what our abilities are, to be able to actually now partner with our scientists, our physicists, our chemists, our biologists, in making these material systems that generate from the subatomic level up to the classical level and beyond.

I'm going back to the first question. I find that we, as artists, have much more of a connection with our scientists, and the hardest cross cultural connection with our engineers. I've had some of my colleagues say, "Why does the media arts program partner with these people? If we're the hardest partner why was it about combining music, digital arts, electrical engineering and computer science?" and I said, "because those are the four basic essentials for media systems design, as hard as our cultural differences are." If you think about it, the scientists like us, have content that they're wrapping their mind around and they're asking questions of discovery that they can't answer right now. Engineers are solving problems constantly. This is what's very difficult about the difference in our cultures. The fact is that we cause problems, we make more questions. Our colleagues are trying to solve those problems and quantify the questions. But if we can partner, then it's going to be easier to partner around our content areas.

The most important thing is why these two content areas — music and digital arts — are the ones that are chosen? Because these content areas are ones that are agnostic to the discipline. We don't care if it's biology or chemistry, we're not going to make an interface if it's discipline-specific. We're going to be the most open in representing information in the best and most cohesive ways. That's what I'm doing with Allosystem. I'm building a general-purpose, visual, aural, multi-user, multi-modal instrument that will look at where the rubber meets the road in every discipline and say "what are the commonalities among those disciplines?" Can we make an all encompassing data discovery tool that's going to sit on top of data analytics and machine learning, that will allow us to represent any information?

I believe we have the path to that.

▲

And speaking of differences, as one of the founders of the department, how would you describe the differences between what was planned when the department was founded and where the department is now?

It's really weird, because for me, what was planned is exactly the direction we're going in. It's funny because most of my colleagues are now just starting to understand this. First of all, it wasn't about big data when I started. Big data didn't even exist. We were going to build an information technology center that would have its first goals of intersecting with its researchers by how they were going to represent their data. Most of our researchers in this building [deal with] data that is really, really small. So for me to get really big, you had to go really small. This whole idea of representation, of visual and aural and interactive representation, immediately brings together the group that you can see. I can outline it. I was very strategic in why I picked and chose the types that I chose. You look at the visual side: you got Marcos [Novak]. Where are Marcos and I? You see the Allosphere, it's an instrument, it's an instrument you fly or drive. You see Marcos representing an environment, a world, that you can navigate, that you can walk through. We're both 3D, we're both immersive, we're both interactive, and we're thinking about visual and audio information. I might be more audio-centric and Marcos might be more visual-centric, but this is where the rubber meets the road. You can now see the connections between the instruments. We've got two different instrument types that have different assets and different limitations, so how are we going to take our content and represent our content differently in these two instruments?

George [Legrady] was primarily working on 2D media that was static and dealing with infoviz. There are two main areas that deal with data representation: infoviz and sciviz. Sciviz mainly deals with volumetric spatial data. Infoviz is more abstract. Yet again, we can pick out the best practices of sciviz and use it in infoviz and vice versa. Then we put Marco on top of this.

Marco [Peljhan] is much more distributed i.e., we are talking about material computing and distributing out the Allosphere into distributed devices and mobile devices. This is what it's all about: you take a big instrument that's all in one place, pull it apart and decide how you are going to re-design the little pieces you have of it and decide what limitations you are going to impose on those pieces.

Think about the areas of research that this defines. I'm at the quantum and sub-atomic level. Marcos has to do with emergence and morphogenesis and is very interested in the biological side. George and Marco are focused on the social and ecological aspects of us as human beings. They are looking at social systems. On the audio side, we are still representing the same thing. We have Curtis [Roads] with his granular and particle synthesis that connects directly with sonification at the quantum level. You can organize his structures into the classical aspect of thing.

We make a community, a very cohesive community here.

I believe in the ground truth principle of science and art coming together. Which means if I'm working at the subatomic level, I should be using density function theory and quantum mechanics. If I'm working at the chemical level, I should be using Monte Carlo simulators, stochastic laws and probability theory.

■

Isn't that a one way street?
The artists that are interested in science eventually manifest this third culture.
But are the scientists interested in acquiring the skills the artists have?

▲

Last quarter we were working on some data visualization pieces with George Legrady (working with the New York Times API). One of your most recent mentions is about a 1995 piece, "*Speira*".

My question is: you have said before that the last 30 years of your life you have been totally devoted to this program. How have you balanced making your own work with this very demanding job?

Let me tell you, that it's a one way street because this is how society has things built. Because of what the scientists right now have to lose due to the formalization of their fields. And that's why we are in here to upset the apple cart. I've been here for 14 years working with physicists and material scientists, and for 30 years with computer scientists and electrical engineers. But only 14 years working with my physicists and materials scientists. And these guys went from not giving me the time of day in 2007, to saying, "You know what JoAnn, we think you're right, but we're not going to go there yet, or let people know out loud that we think you're right because we've got too much to lose." A lot of people think it's stupid, they think it's backward. But it's also where I am in my 30 years of making and I don't expect all my students to be in this same place. No, you don't have to follow the exact ground-truth principles that I'm talking about and you don't have to be valid to scientists. You're artists for heavens sake! You can take from them what you want and you can make whatever you want to make.

For me, I've been doing that for 30 years. It's very important at this point that whatever I do follows physical phenomena. I've lost the ability to make because I've lost the technical ability to make what I want to make. So it is a one way street now, but I hope I'm paving the way for you guys to make it a two way street. And not only is it going to be a two-way street, they're going to be knocking on the door.

Oh right, *Speira*.

The thing is this is a question of balance. It's a question of balance that I ask my colleagues at all times. We all have to make the decisions that we make personally. I've made a different personal decision that a lot of my colleagues have. Before I started building the community in 1994 I was very famous as a composer. I won every competition, I was in every computer music conference, I was invited to every venue. I did the whole spiel. From the 1980s until about 1995. 1995 was when I became Associate Dean for computing. But for me it was never a break away from my original plan. In 1984 when I wrote my string orchestra piece, which was *Tachyons*, it was about the stellar collapse of the universe, it was about a particle that moved faster than the speed of light, there was a black hole, white hole effect. It was a big kind of Ligeti's *Atmosphères* piece. All done with equal-tempered tuning, all done with set and group theory. And yet again, the story that I was telling was not the math that I was using. And I always wanted to be true to what my visions were. This has been a 30 year push for me to build, to selfishly build, the instrument that I want to build that's going to allow me to do what I want to do. Unfortunately this has pushed me to a different place for creation and I have to come to terms with that. As I say there are innovators and culminators in every field. You look at J.S. Bach and you look at C.P.E. Bach and you see a culminator – J.S., and you see an innovator — C.P.E. Unfortunately I'm on the innovation side, and unfortunately there are composers, you can look over the history, most of these people go through hiatuses of creation.

They are changing their style, they are changing their vocabulary and they don't have the tools that they typically had in order to create, and I'm one of those people, I'm completely in limbo right now.

I can sit down and I can write any piece in few seconds and I can write anything that deal with equal-tempered, micro-tonal grammars like you wouldn't believe, that's where I come from. But pieces that I want to write I don't have the mathematical ability or the technical ability to do that, so I have to work with teams.

■

Actually I think this consequence that you're drawing — to not write a piece because you don't have the math to do it — is the completely right path. So many people that create will not stop, they will just pretend. They will work with representations and metaphors without looking at the actual structures.

I have to do that. And if I ever make another piece, I will make a piece on the hydrogen-like atom. I will perform that piece it won't be completely real, the only way that I can actually make a probability wave function move into an inordinate number of them is through a mathematical perturbations that Luca [Peliti] has to figure out on the Schröedinger equation and he said it's gonna take him another couple of years and I´m running out of money. I'll still make a piece on that one.

It really won't be completely mine. That piece is gonna performed and we'll have Lance and Luca and myself equally on that line and I'll try to do something and maybe that will come out and they might want to premiere at the opening of Moxi, the museum downtown in 2016. Which is about right.

2016, 2017 maybe I'll come out with this work but it still won't be right, but it'll be nice. You have to lose your ego, you have to just get rid of the "I" completely.

END OF YEAR SHOW 2015

LIST OF **WORKS**

PALETTE

Anis Haron

This project is a series of algorithmically generated prints. I developed a program that uses the Google custom search API to generate multiple permutations of a set of colors. After deciding on a search term (ie. summer), the program selects 10 images retrieved by the API. The images are selected from the search results in a sequence starting from a random position, and average color is calculated for each of the 10 images and its values are stored. A second array contains color values from the average colors sorted by hue, while a third array re-sorts the colors in a configuration relative to the first array. These three sets of color sequences are graphically represented in equal sided squares drawn in every possible permutation.

/ anisharon.com

THE BODY IS PRESENT?

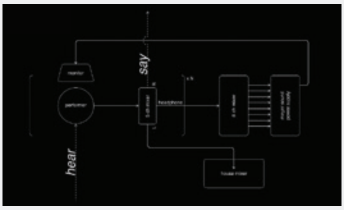
Sahar Sajadieh

The Body is Present? is a durational participatory performance piece, which is a digital reperformance of Marina Abramovich's *The Artist is Present*, performed in 2011 at the Museum of Modern Art in New York City. This performance will problematize and provoke dialogue about the nature of virtual interaction, presence, and embodiment in telecommunication. The piece is a theatrical Turing test, examining the spectator's perception towards the presence of the other body on the other side of the digital medium.

/ saharsajadieh.wordpress.com

ENSEMBLE FEEDBACK

CREATE Ensemble



Ensemble Feedback is a 15 minute structured musical improvisation. 5-8 players excite and control a sparsely-connected feedback delay network, effecting changes in the graph topology and adapting to those changes in real-time.

Each player's instrument interfaces with the feedback delay network via a matrix mixer. The output of each instrument is also connected to a monitor and speaker creating a 5-8 channel surround work.

/ create.ucsb.edu/ensemble

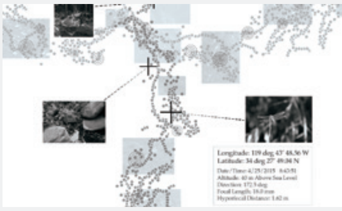
TIBETAN SINGING PRAYER WHEEL

Cecilia Wu · Yoo Hsiu Yeh

Tibetan Singing Prayer Wheel is a hand-held, wireless, sensor-based musical instrument with a three-dimensional human-computer interface that simultaneously controls various interactive electronic voice processing and sound synthesis techniques through each single gesture. Inspired by both a traditional spiritual instrument, the prayer wheel, and a traditional acoustic musical instrument, the Tibetan singing bowl, this hybrid instrument contains miniaturized electronics, sensors, a wireless transmitter and real-time software that processes vocals and synthesizes sound based on the performer's hand gestures. A one-to-many mapping strategy enhances the performer's control and musical expression. A physical model implementation simulates the traditional Tibetan singing bowl. Modal synthesis as well as delayed and windowed processing process the performer's vocals. This system is designed for an electroacoustic vocalist interested in using a solo instrument with a single hand to achieve multiple performance goals that would normally require multiple instruments and activities.

OPEN SPACES

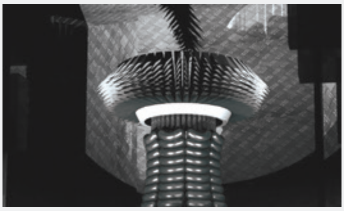
David Gordon


Longitude: 119 deg 43' 48.56 W
Latitude: 34 deg 27' 49.04 N
Date/Time 4/25/2015 8:43:51
Altitude: 40 m Above Sea Level
Direction: 172.5 deg
Focal Length: 18.0 mm
Hypothetical Distance: 1.62 m

Open Spaces is a two-part media artwork created from over 15,000 digital photographs and hundreds of field recordings taken in three local Santa Barbara parks: Tucker's Grove, Rattlesnake Canyon and Jesusita Trail. By linking together images and sounds with their spatial locations, the project investigates the use of technology to create a comprehensive picture of a place. *Open Spaces* developed out of a longtime interest in the natural environment, the practice of deep listening, and the use of photography to capture details in a space that are often overlooked due to the distractions and fast pace of contemporary life. *Open Spaces*: Pathways, is an interactive audiovisual installation consisting of a virtual 3D simulation made from photos and recordings of these three nature areas. Photos and field recordings were taken at roughly evenly spaced points along trails in each park. For each media file, geographical location, altitude, and compass direction were recorded using a GPS device. The software loads this data and uses it to spatialize each media file in the virtual environment. As viewers interact with the installation, they see images and hear sounds around them in the approximate locations where they would appear in the real world. Images and sounds fade in and out as virtual time moves from dawn to dusk over several minutes, allowing the viewer to experience a complete "day" in each of the three parks. *Open Spaces*: Transitions is a seven minute panoramic video created for the Allosphere. The work consists of 360-degree panoramic images taken in each of the three parks. These panoramas were ordered according to geographical proximity and blended together, creating a continuous visual transition that reflects a gradual spatial movement through the environment. Sound recordings taken in the corresponding locations were time-stretched and blended with each other to create a gradual sound transition analogous to this visual process.

CORAL TREE

Intae Hwang



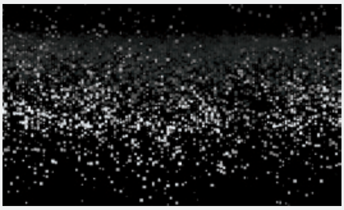
Media Arts and Technology and the California NanoSystems Institute are located in Elings hall. It is known that one of the labs utilizes a nuclear reactor. Living with this potential hazard increases curiosity and anxiety. I transformed this building into a futuristic one. The colorful doors are activated by the player's movement. Deep inside the building in the game, there is an object which is similar to the nuclear reactor. The blue color addresses the feeling of being somewhere inside a hidden military facility. To create the object, I applied several techniques in Rhino (using a combination of gloss, transparency, reflectivity, and lighting effects). A dark indoor space and shadow effects enhance the atmosphere. Experiencing immersive architectural space was one of the goals of this project. This topic was explored by Romantic painters in 19th century. Caspar David Friedrich addresses this emotions in his paintings. Standing in front of enormous nature, humans are being reduced to their small existence. This project allows players to explore futuristic cubic places, which eventually turn out to be small parts of a gigantic complex of mega buildings.

/ intaehwang.net/

LU:.KI.DUS

Kurt Kaminski • Juan Manuel Escalante

ENGINEERING ADVISOR: Akshay Cadambi

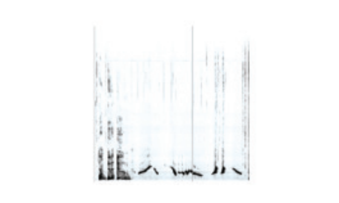


Lu:.ki.dus is a live cinema and sound performance focusing on scale, pattern, and awareness. We developed a technique to automate a microscope's stage position through computer controlled motors, allowing for both complex orchestrated maneuvers as well as dynamic performance and viewer interaction. In addition to microscopic imagery, the project simultaneously brings in live video data from other sources for alternative perspectives. Algorithms distort and enhance the video as the performance evolves. A five movement electronic music soundtrack was composed exclusively for the performance. Our goals revolve around exposing the unseen yet omnipresent events which unfold at the microscopic scale. We will be crafting a distinctive kind of microscopic world that leans more towards the experimental rather than scientific. We feel that prolonged viewing of vibrant, real-time microscopy has a noticeable impact on the audience's perception and appreciation for the natural world. Scenarios photographed and included in the performance are: local pond water containing microscopic organisms, crystal growth, fluid dynamics via paints and solvents, and unusual found-materials such as skin care products

/ kurtkaminski.com / realitat.com

SIMPLE THINGS

Tim Wood



This performance is a live reading of spoken word poetry in combination with audio looping and various acoustic instruments. The project is a playful improvisation of my poetry, utilizing digital audio looping as a method of repetition in order to generate an alternative multi-dimensional reading experience.

/ fishuyo.com

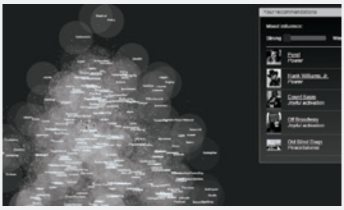
FOUR EYES LAB

DIRECTED BY MATTHEW **TURK** • TOBIAS **HÖLLERER**

MOODPLAY

Ivana Andjelkovic

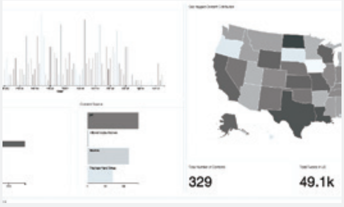
Acknowledgments: John O'Donovan (UCSB, Computer Science), Shlomo Dubnov (UCSD, Music), Denis Parra (PUC Chile, Computer Science), Curtis Roads (UCSB, MAT)



MoodPlay is an interactive and adaptive mood based music recommender. The development of this system has been motivated by a need to explore music collections by exhaustive list of words we use to describe moods. Music has a tremendous effect on our emotions, and finding artists whose music feels right at a given moment is not always easy when we face millions of choices. To address this problem, *MoodPlay* organizes over 250 moods into a hierarchy and visualizes them in a novel way. The interactive interface built upon proposed mood model and hierarchical mood filtering make finding relevant artists easier. Furthermore, the system implements a novel, hybrid recommendation algorithm, which adapts to mood changes that naturally occur while listening to music. The user study conducted on 216 subjects showed that the proposed model and algorithm improve user satisfaction and recommendation accuracy.

STREAMSCOPE: SOCIAL STREAM ANALYZER

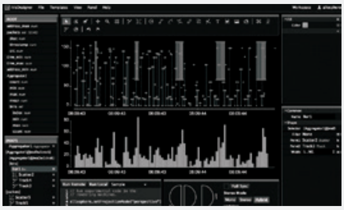
Byungkyu (Jay) Kang • John O'Donovan



StreamScope is an interactive web-based visualization that represents a real-time landscape of topic-specific conversations and real-world trending events on social media streams. Since today's web technology has changed the paradigm of communication to one with vast amounts of user-generated content, it is increasingly important to improve and adapt our information filtering tools. StreamScope achieves this by applying state-of-the-art filtering algorithms to streaming content in real time in an easy-to-use interface. This framework is designed to help both data analysts and novice users gain insights into social media data. In addition to the interactive tool itself, the key research contribution of this work is an exploration of the limitations and synergies that exist between human information analysts and interactive, but largely automated information filtering algorithms.

IVISDESIGNER: EXPRESSIVE AND INTERACTIVE DESIGN OF INFORMATION VISUALIZATIONS

Donghao Ren

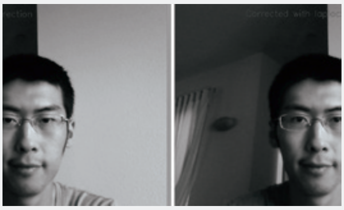


iVisDesigner stands for Information Visualization Designer, which is a platform in which users can interactively create customized information visualization designs for a wide range of data structures.

donghaoren.org

EYE GAZE CORRECTION

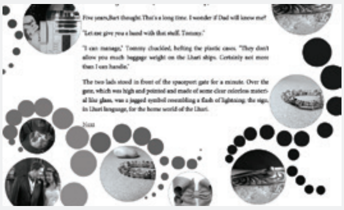
Kuo-Chin Lien · Yalun Qin



In traditional video conferencing systems, it is impossible for users to have eye contact when looking at the conversation partner's face displayed on the screen, due to the disparity between the locations of the camera and the screen. In our project, we present a gaze correction system that can automatically maintain the eye contact by using computer vision and computer graphics technology. Our system does not require any hardware apart from a web-camera, and supports the camera in almost any location around the laptop.

SIDESTORIES

Carrie Segal



Five years,Bart thought.That's a long time. I wonder if Dad will know me?

"Let me give you a hand with that stuff, Tommy."

"I can manage," Tommy chuckled, hefting the plastic cases. "They don't allow you much baggage weight on the Lhari ships. Certainly not more than I can handle."

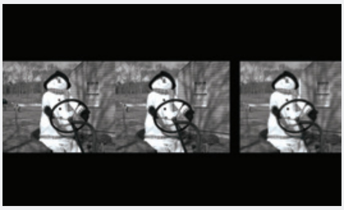
The two lads stood in front of the spaceport gate for a minute. Over the gate, which was high and pointed and made of some clear colorless material like glass, was a jagged symbol resembling a flash of lightning; the sign, in Lhari language, for the home world of the Lhari.

Next

SideStories is a story illustration algorithm which places annectent images alongside the authored pages of text. Colors and adjective-noun pairs are used to find words representing things, and a one-of-a-kind handmade website is queried to find photographs. The completed illustration is a digital book page with a border of images and colors.

SEAM CARVING PROJECT

Kuo-Chin Lien



The major objective of image retargeting algorithms is to preserve the viewer's perception while adjusting the size or aspect ratio of an image. This means that an ideal retargeting algorithm has to be able to preserve high-level semantics and avoid generating low-level image distortion. Stereoscopic image retargeting is a more challenging problem in that the 3D perception has to be preserved as well. We present a stereo pair retargeting algorithm to simultaneously guarantee the 2D as well as 3D quality.

[IN]FLAME

Nataly Moreno · Mohit Hingorani

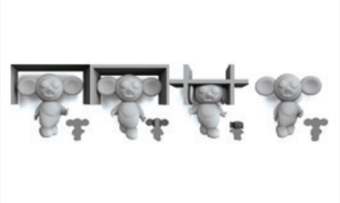
A mobile application for the iPhone 5 using the FLIR infrared camera to aid in sports medicine by detecting inflamed joints. Inflamed joints tend to be about one degree hotter than joints that are not injured. The application guides the user to photograph the patient and it compares the symmetrical body parts to detect inflammation.

MIRAGE LAB

DIRECTED BY THEODORE **KIM** · PRADEEP **SEN**

SUBSPACE CONDENSATION

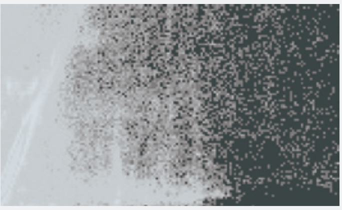
Yun Teng · Mark Meyer · Tony DeRose · Theodore Kim



Subspace deformable body simulations can be very fast, but can behave unrealistically when behaviors outside the prescribed subspace, such as novel external collisions, are encountered. We address this limitation by presenting a fast, flexible new method that allows full space computation to be activated in the neighborhood of novel events while the rest of the body still computes in a subspace. We achieve this using a method we call subspace condensation, a variant on the classic static condensation precomputation. However, instead of a precomputation, we use the speed of subspace methods to perform the condensation at every frame. This approach allows the full space regions to be specified arbitrarily at runtime, and forms a natural two-way coupling with the subspace regions. While condensation is usually only applicable to linear materials, the speed of our technique enables its application to non-linear materials as well. We show the effectiveness of our approach by applying it to a variety of articulated character scenarios.

ROBUST PATCH-BASED HDR RECONSTRUCTION OF DYNAMIC SCENES

Pradeep Sen · Nima Khademi Kalantari
Maziar Yaesoubi · Soheil Darabi · Dan B Goldman
Eli Shechtman

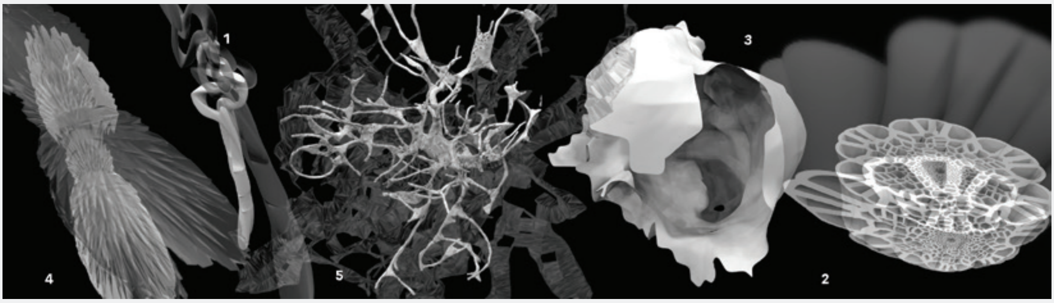


High dynamic range (HDR) imaging from a set of sequential exposures is an easy way to capture high-quality images of static scenes, but suffers from artifacts for scenes with significant motion. In this project, we propose a new approach to HDR reconstruction that draws information from all the exposures but is more robust to camera/scene motion than previous techniques. Our algorithm is based on a novel patch-based energy-minimization formulation that integrates alignment and reconstruction in a joint optimization through an equation we call the HDR image synthesis equation. This allows us to produce an HDR result that is aligned to one of the exposures yet contains information from all of them. We present results that show considerable improvement over previous approaches.

PATCHBASED HIGH DYNAMIC RANGE VIDEO

Nima Khademi Kalantari · Eli Shechtman · Connelly Barnes · Soheil Darabi · Dan B Goldman · Pradeep Sen

Despite significant progress in high dynamic range (HDR) imaging over the years, it is still difficult to capture high-quality HDR video with a conventional, off-the-shelf camera. The most practical way to do this is to capture alternating exposures for every LDR frame and then use an alignment method based on optical flow to register the exposures together. However, this results in objectionable artifacts whenever there is complex motion and optical flow fails. To address this problem, we propose a new approach for HDR reconstruction from alternating exposure video sequences that combines the advantages of optical flow and recently introduced patch-based synthesis for HDR images. We use patch-based synthesis to enforce similarity between adjacent frames, increasing temporal continuity. To synthesize visually plausible solutions, we enforce constraints from motion estimation coupled with a search window map that guides the patch-based synthesis. This results in a novel reconstruction algorithm that can produce high-quality HDR videos with a standard camera. Furthermore, our method is able to synthesize plausible texture and motion in fast-moving regions, where either patch-based synthesis or optical flow alone would exhibit artifacts. We present results of our reconstructed HDR video sequences that are superior to those produced by current approaches.



MORPHOGENESIS WORKSHOP

DIRECTED BY

JEAN-MICHEL **CRETTAZ**

MORPHOGENESIS

This course introduces the conversation of physical forms and their material manifestations to a students ongoing research topic. Through a period of 10 weeks, students are asked to select a particular process within their ongoing research and engage in possible ways that this process is materialized in a tangible 3D object. The exploration of how ideas can generate the process generating possible forms through selected computational processes is a particular interest in the development of how to engage with our physical world. This course is designed to be a catalyst for such questions and through the weekly discussions and desk-crits each students conversation expands to incorporate the methods of dynamic physical processes. The integration of subdivision as a method and thought process is presented as a framework that embraces the continued development of unique design and thought processes. Throughout the investigative process reflection is a key component, contributing to how the initial ideas are redefined in order to focus the questions being asked and the methods being used for the further investigation and continued conversation. The fabrication of a 3D printed object with corresponding drawings and diagrams is the result of each students particular conversation, research and development.

1 · Alexis Crawshaw · Caduceus
These two staves— which collectively evoke the caduceus of Hermes, Greek God of continua and boundaries— are designed as spatial-tracking objects that serve to create a gestalt musical experience in relation both to each other and across other sensory modalities within a transmedia performance: Trans-Sensory Music.

2 · Intae Hwang · Mandelbrot plant
This object started with a small part of the Mandelbrot set fractal. The pattern grows with the cell fusion reaction. From 400 cells generated by the Mandelbrot set, a single cell remains when the growth is done.

3 · Gustavo Alfonso Rincon · A Repulsive Embrace of a Flock of Emotions
The form is an embodiment of a series of experiments and the capturing of a moment in time of a flock of agents responding to stimuli.

4 · Hannah Wolfe · Bodily Extensions
An exploration of form and embodiment for ROVER, an interactive sculpture, using fluid vector fields, the Catmull-Clark algorithm, spherical harmonics and Michael Hansmeyer's "Design by Subdivision" algorithm.

5 · Tim Wood · Kinetrope
Kinetrope is a plant-like lifeform born in virtual space, that grows towards the motions of a body in physical reality, and crosses the boundary of the virtual to meet the body face to face.

MAT Media Arts and Technology Graduate Program

Designed at 3320 Phelps Hall | University of California, Santa Barbara | MMXV