Recognizing Creative Visual Design: Multiscale Design Characteristics in Free-Form Web Curation Documents

Ajit Jain[1], Andruid Kerne[1], Nic Lupfer[1], Gabriel Britain[2], Aaron Perrine[1]
Yoonsuck Choe[3], John Keyser[3], Ruihong Huang[3]
[1]Interface Ecology Lab, Dept. of Computer Science and Engineering, Texas A&M University
[2]Georgia Tech, [3]Dept. of Computer Science and Engineering, Texas A&M University
{ajit,andruid,nic}@ecologylab.net,quontas@gmail.com,avsp.here@gmail.com,{choe,keyser,huangrh}@cse.tamu.edu

ABSTRACT

Multiscale design is the widely practiced use of space and scale to visually explore and articulate relationships. Free-form web curation (FFWC) is an approach to supporting multiscale design, involving creative strategies of collecting content, assembling it to juxtapose and organize, sketching, writing, shifting perspective to navigate, and exhibiting to share and collaborate. Our long term goal is to support design students with automatic, on demand feedback.

We introduce a spatial clustering technique for recognizing multiscale design characteristics—scales and clusters—in FFWC documents. We perform quantitative evaluation to establish baseline performance. We contribute to human-centered AI by advancing fundamental human aspirations, through automatic recognizers of creative design, e.g., for representing and communicating abstract ideas. We develop implications, (1) for supporting people using content recognition in creative contexts, such as design education; (2) for overcoming design fixation with human-centered AI; and (3) for recognizing multiscale design characteristics.

CCS CONCEPTS

• **Human-centered computing**; • **Applied computing** → **Education**; • **Computing methodologies** → *Cluster analysis*;

KEYWORDS

design education, curation, multiscale design, spatial clustering

1 INTRODUCTION

To support design education, we introduce an approach to recognizing multiscale visual design, a characteristic of what Tufte calls *escaping the flatland of envisioning information*, i.e., using design

strategies to increase the legible dimensionality and density of information on the screen and page [52]. Lupfer et al. define *multiscale design* as, "The use of space and scale to explore and articulate relationships, [which] involve the juxtaposition and synthesis of diverse design elements" [35]. Systems supporting multiscale design—e.g., Photoshop, Illustrator, InDesign, and IdeaMâché [36]—enable extending basic two dimensional design by organizing content across a range of scales. When doing this work with interactive computing systems, users change focus to traverse zoom levels. An investigation of a landscape architecture classroom found that multiscale design pervades student work on projects [35]. In these projects, multiscale design supports students in schematically forming design proposals to meet the situated needs of sites involving waterways and land use. In a study involving computer science courses, multiscale design has been found to support students' iterative and reflective ideation [36].

We have been conducting an extended field study in design education courses. In these contexts, students engage in multiscale design through *free-form web curation* (FFWC)—a form of new media—which involves collecting and assembling digital artifacts to create new spatial and conceptual document structures (See Figure 1) [25, 36]. From our observation, as well as prior work [43], we find that students engaging in design work need frequent, helpful feedback to make progress.

Consider a scenario. Sue, a novice student, works late on a conceptual design assignment, the night before it is due. At this time, the course instructor is unavailable to provide feedback. Sue's design gets stuck in flatland: many content elements are presented, but without spatial organization that discernibly depicts categories and other relationships. Sue would benefit from computational recognition of characteristics of multiscale design in their FFWC document, in order to support them in reflecting on and improving their design work. The multiscale design recognition algorithm developed in the present research would help Sue understand their current status. With additional research, it can be coupled with hints for how to better organize content elements, to escape flatland.

Prior techniques for analyzing spatial documents—e.g., snapshots of articles and magazines, and other image files and PDFs—support various uses, e.g., determining whether the role of visual content is structural or illustrative [14], correcting errors based on content placement [8], and creating document descriptors for authentication [9]. In education contexts, computational recognition of spatial and visual content has supported student learning, such as by automatically generating models from engineering diagrams [13] and providing recommendations based on lifelog images [20]. However, prior work did not focus on recognition of how designers use space
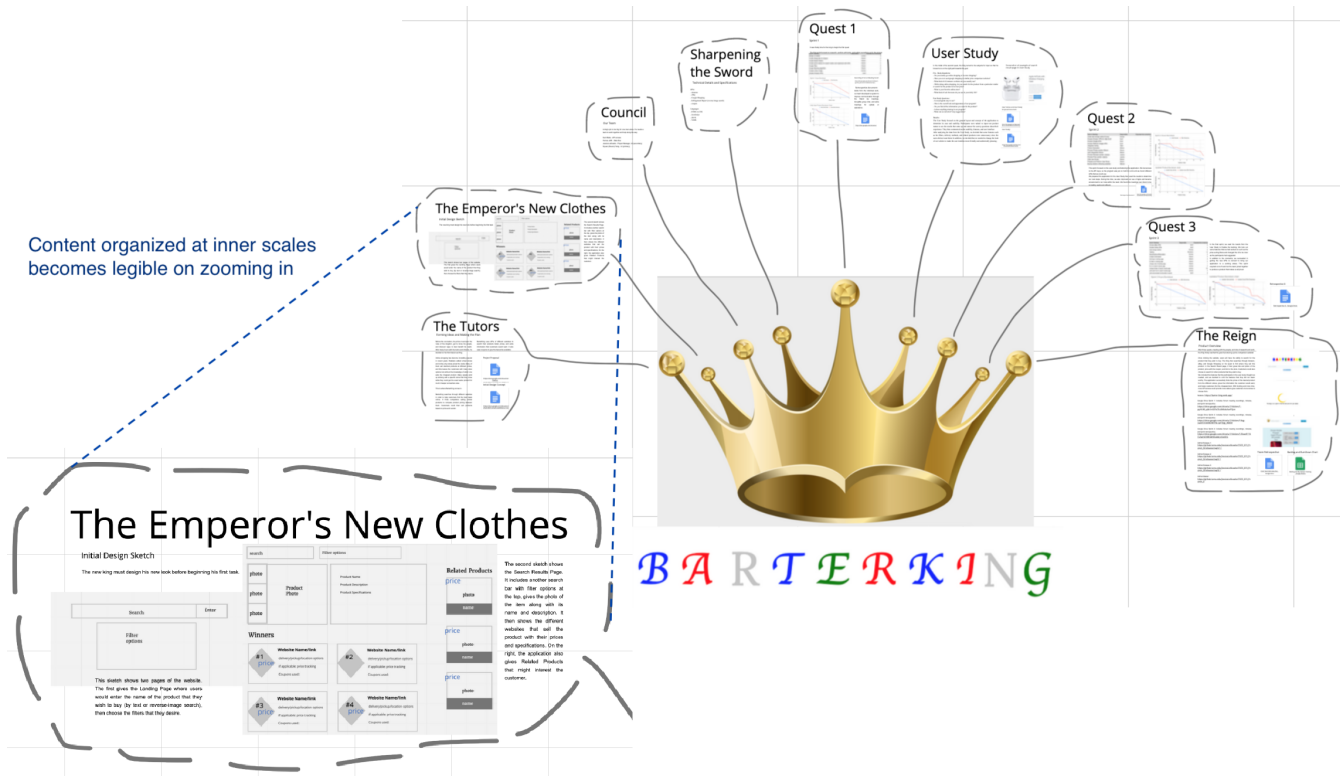
Figure 1: A student's multiscale design for instructor I4's human-computer interaction project. They organize phases of the project as design elements, nested across scales of magnification. At the outer scale, a crown, its caption, spikes connected with each phase, and headings within each phase together form a holistic cluster. Bottom left, see the view as the user zooms in. Elements deeply nested in one scale become legible in the next. We observe nine nested clusters at the inner scale.

and scale to represent concepts involving complex, extended sets of content elements. It also did not address free-form web curation as a document structure to support graphical, visual expression.

The present research addresses the gap, to support people in going beyond flatlands of design to represent complex collections and ideas across levels of scale. We advance investigation of multiscale design characteristics recognition, in order to provide instructors and students with insights about design work. We next present prior work for computational design recognition through spatial clustering. We follow with the FFWC document structure. Subsequently, we present the multiscale design characteristics recognizer algorithm, extending a prior spatial clustering technique, along with its performance validation. We discuss implications for computational recognition of creative visual design, focusing on multiscale characteristics. We consider the potential for this approach to support design education.

2 PRIOR WORK

We present prior work on spatial clustering relevant to recognizing multiscale design characteristics. Spatial clustering refers to the partitioning of "spatial data into a series of meaningful subclasses called spatial clusters, such that [elements] in the same cluster are similar to each other, and are dissimilar to those in different clusters" [34]. Multiscale design is characterized by the organization of content across zoom levels, which typically results in nested

clusters (Figure 1). Taking inspiration from recent success on the use of clustering for modeling design in 2D space [51], we investigate the potential of algorithms that identify clusters within clusters, for the purpose of recognizing multiscale design characteristics.

Liu et al. and Deng et al. classify spatial clustering algorithms into seven groups: partitioning, hierarchical, density-based, graph-based, grid-based, model-based, and combinatorial [7, 34]. They discuss that most of these algorithms require the number of clusters as an input. In design contexts—where differences in the representation of ideas are a rule rather than an exception [16]—the number of clusters is not fixed in advance.

Out of the four algorithms that do not need the number of clusters as an input, only graph-based AMOEBA [11] and AUTOCLUST [10] show promise for this research. AUTOCLUST is an extension of AMOEBA, with additional criteria addressing adjacency of sparse and high-density clusters. Both algorithms take a Delaunay triangulation based approach to determine the nested clusters. The algorithms were previously utilized in the application domain of geographical information systems—with locations input as points—allowing investigating phenomena such as earthquakes, land use, and customer spread [7]. However, the algorithms do not lend themselves to the application domain of design education. FFWC documents in our design education contexts consist of 2D visual design elements—such as text, image, and video—which

are characterized by a region, and not just a point of interest. The present research addresses the challenge of adapting the Delaunay triangulation based approach for design contexts, with 2D elements.

3 FREE-FORM WEB CURATION DOCUMENTS

Building on art practice, free-form web curation (FFWC) is defined as, "a form of new media—designed to support users in creating new conceptual and spatial contexts—in which they discover, interpret, and represent relationships, by composing readymade and self-made content elements, on the web and in the cloud" [25]. *Readymade* here refers to postmodernism's 'found objects'—from Dada [33] to conceptual art [2] and beyond—of multimedia content, such as text, images, audio, and video—that users collect from diverse sources—e.g. news, social media, and e-commerce web pages—and integrate them in the new context of a curation. By an *element* then, formally, we mean content that is in this sense readymade, and so now collected by an "author", or self-made, i.e., created by them.

An FFWC document (Figure 2) formally consists of a collection (2.a) of content elements (2.b), each with graphical transformations (2.c). The document level (2.a) also stores properties, including title, description, key (used in web URL), id (a unique internal identifier), settings (visibility and background color), and creator. The FFWC system of the present investigation supports the collection, assemblage, rendering, and storage of content elements—on the web and in the cloud—through the use of HTML, Javascript, and CSS web technologies. FFWC documents are stored in a database as JSON.

In developing and navigating an FFWC document, designers invoke creative strategies of: collect, assemble, sketch, write, shift perspective, and exhibit [25]. Using the FFWC system, they *collect* content elements through drag 'n' drop from different web pages. They position, scale, and rotate elements to organize them visually. As part of how the system performs the collect strategy, in response to the user, a Chrome Browser Extension extracts semantics of the source web page, which comprises at least the page title and URL [46]. The system associates the extracted semantics with a reference to the collected element, which is together referred to as the *clipping* within an element (Figure 2, label (d)). Depending on the source of clipping—e.g., news article, social media post, or scholarly article—the semantics may contain additional information. For example, semantics for a scholarly article include additional information, such as authors, references, and citations. Further, each element clipping, depending on the content, is assigned a type, such as text, image, sketch, and video.

In developing new conceptual and spatial contexts, designers creatively *assemble*, readymade collected elements, as well as self-made "annotations" [38] that they create by invoking the *sketch* and *write* strategies. Invoking the assemble strategy, that is, organizing content elements and designing the representation of the whole, they perform operations such as move, resize, and rotate, which the FFWC system stores as a set of *transforms* with each corresponding element (Figure 2, label (c)). Self-made sketch annotations are stored in the form of strokes within the respective element's clipping. Self-made writing annotations are stored as characters with font specs. Designers invoke the *shift perspective* strategy, through pan and zoom, in order to navigate across space and scale. The unique URL assigned to each FFWC document is used to support

```
{
  title: 'Project Sketch Mache',                          (a) FFWC Document
  description: 'Milestone 1',
  elements:                                               (b) Elements
  [ { transforms:                            (c) Transforms
      { scale: { x: 13.888, y: 13.888 },
        position: { z: 45, x: 450.363, y: 96.496 },
        rotation: 0 },
    opacity: 1,
    anchored: false,
    _id: 5dc4e149ceae892ec404abee,
    key: 'BtyPw_nOT',
    creator: 5d8db71843d90b24d876954c,
    parent: 5dc4de7cceae892ec404ab23,
    createdAt: 2019-11-08T03:30:17.309Z,
    updatedAt: 2019-12-23T05:28:01.866Z,
    clipping:                                             (d) Clippings
      { semantics:
        { primaryLocation:
            'https://www.youtube.com/watch?v=1s-q5oy5lhc' },
          videoClipping: { provider: 'youtube.com', videoId: '1s-q5oy5lhc' },
          _id: 5dffdc4c92e4456283285aa9,
          clippingType: 'videoClipping',
          width: 516,
          height: 411,
          parent: 5dc4e149ceae892ec404abee } },
  { transforms:
      { scale: { x: 11.158, y: 11.581 },
        position: { z: 42, x: 112.546, y: 636.349 },
        rotation: 0 },
    opacity: 1,
    anchored: false,
    ...............,
    clipping: ............... },
  ...............
  ]
  settings: {
      visibility: 'public',
      backgroundColor: null
  },
  tags: [],
  _id: 5dc4de7cceae892ec404ab23,
  key: 'EMICPT9sE',
  creator: 5d8db71843d90b24d876954c,
  createdAt: 2019-11-08T03:18:20.945Z,
  updatedAt: 2019-12-23T08:08:08.457Z,
  ..............
}
```

Figure 2: An FFWC document is comprised of properties such as *title*, *description*, *creator*, and a collection of *elements*. Each element includes a *transforms* property—with sub-properties position, scale, and rotation—which allows determining the element's spatial location with respect to the origin. Each element also stores a *clipping* property, which in turn stores semantics extracted at the time of collecting the element.

the *exhibit* strategy. The unique part of the URL is stored as the *key* property within the FFWC document. The *visibility* property value (public or private)—a sub-property of an FFWC document's *settings* property—allows the creator to control with whom they exhibit their work, i.e., sharing and collaboration permissions.

For graphical transforms, the FFWC system uses two coordinate spaces: curation and screen. The curation space extends infinitely in the X, Y, and Z dimensions. The screen coordinate space is only 2D. Its extents depend on display resolution. Applying the transforms stored within an element determine its curation space position with respect to the origin, as well as size and orientation. When

Table 1: Five design course professors and four teaching assistants, in the role of lab instructor, engaged students in creative visual design through free-form web curation.

ID	Gender	Course	Field	Role	Semester
I1	F	Digital Media Design, UI/UX for Games	Interactive Art & Design	Course Instructor	Spring 20
I2	F	Interaction Design	Interactive Art & Design	Course Instructor	Spring 20
I3	F	Engineering Design	Mechanical Engineering	Course Instructor	Summer 20
I4	M	Programming Studio	Computer Science and Engineering	Course Instructor	Spring 20
I5	M	Programming Studio	Computer Science and Engineering	Course Instructor	Spring 20
I6	F	Programming Studio	Computer Science and Engineering	Lab Instructor / TA	Spring 20
I7	M	Programming Studio	Computer Science and Engineering	Lab Instructor / TA	Spring 20
I8	M	Programming Studio	Computer Science and Engineering	Lab Instructor / TA	Spring 20
I9	M	Programming Studio	Computer Science and Engineering	Lab Instructor / TA	Spring 20

rendering an FFWC document, a curation to screen coordinate space mapping is performed—applying scale transforms based on the ratio of screen space and curation space dimensions—to render each element within its screen extents. The Z dimension similarly specifies resizing elements through additional scale transforms, at rendering time. Hence, if two identical elements are assembled at two different zoom levels, the one at the inner level will appear smaller. The semantic information associated with an element is only rendered on demand, when the user requests it, via the context menu presented on right-click.

4 RECOGNIZING MULTISCALE DESIGN CHARACTERISTICS

The present research introduces investigation of computational recognition of creative design work, i.e., how design is multiscale, through analysis of FFWC documents. Based on what multiscale design comprises, the characteristics we recognize are the numbers of scales and clusters. *Scale* refers to levels of magnification: elements at the same scale are comparably legible at the same viewport zoom [36]. The juxtaposition and synthesis of elements, across scales, can produce nested spatial groups, or *clusters* (Figure 1).

4.1 Design Curation Contexts

We investigate learning contexts in which students are expected to engage in multiscale design through creating FFWC documents. We ran a preliminary field study[1], in which nine teaching team members in five courses (Table 1) had students engage in creative visual design through free-form web curation. The courses included: Digital Media Design, UI/UX for Games, and Interaction Design in the Department of Interactive Art & Design; Engineering Design in Mechanical Engineering; and Programming Studio in Computer Science and Engineering. A total number of 235 design curations were created by students in these courses.

4.2 Recognizer Algorithm

Based on our survey of prior spatial clustering algorithms (Section 2), we chose the AMOEBA recognizer [11] because: (1) it models spatially nested clusters, and (2) it does not require the number of clusters as input. Designers freely vary the number of clusters. The algorithm is computationally efficient, requiring *O(nlogn)* time.

AMOEBA has previously been used for point elements. It uses Delaunay triangulation to recursively determine nested clusters,

based on distances among elements. Delaunay triangulation [32] refers to the formation of triangles by connecting a given set of points such that no points lie inside the circumcircle of any triangle. The Delaunay graph is formed by connecting all triangle vertices through edges. AMOEBA compares the lengths of edges, which are incident on each element vertex, with the average global edge length among all connected vertices in the Delaunay graph. It uses a single hyperparameter α (alpha) to calculate a threshold length. Any edges longer than the threshold are removed. This forms the first set of clusters, i.e., groups of elements with edges only amongst themselves. The hyperparameter α thus controls the homogeneity/heterogeneity of clusters: the bigger the value of this hyperparameter, the more heterogeneous distances exist amongst elements in a cluster. The process is recursively repeated for each resulting cluster until no edges get removed in a recursion step.

We extended AMOEBA to support two dimensional design elements. We iterate over the element collection—stored in the top level of the FFWC document (Section 3, Figure 2)—applying the transforms for each element. This yields the coordinates of each element in the curation coordinate space (We do not need to convert them to screen coordinate space, as the algorithm depends on distances among elements, which are proportional in both coordinate spaces). We create a mapping between each element and its transformed coordinates. We input this mapping into the recognition algorithm. The mapping enables the new algorithm to process elements as spatial regions both for Delaunay triangulation and for calculating proximate element distances. Algorithm steps involved in extending AMOEBA are (See visual explanation in Figure 3):

(a) When computing the Delaunay graph, incorporate the dimensions of each FFWC element via its four vertices. If we only consider the center, then elements having large width/height will be incorrectly separated from those nearby.

(b) When iterating through the edges of the Delaunay graph for computing distances (and comparing with the threshold value), instead of directly using the edge length, find the shortest distance between the two FFWC elements connected by each edge (e.g., d2 in Figure 3). Delaunay triangulation forms edges with nearby elements, but the edges in the Delaunay graph may not always be the shortest distance.

(c) Relatedly, ignore any edges between the vertices of the same FFWC element. The distance of an element to itself is zero.

(d) When calculating subgraphs, collapse vertices of each FFWC element by adding edges among all its vertices. Without such intra-element edges, two vertices of the same element having

---

[1]We obtained IRB approval prior to engaging human subjects in this research.
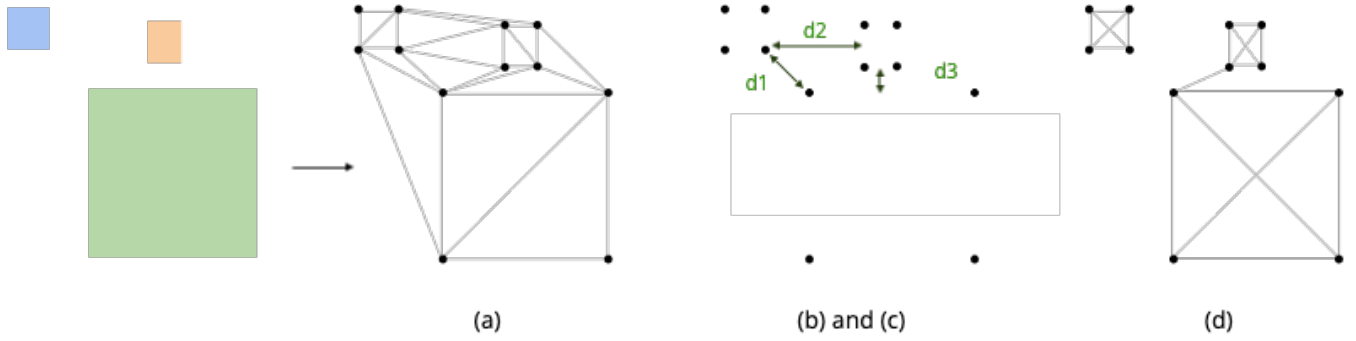
Figure 3: Visual explanation of extending AMOEBA for 2D elements (with width and height): (a) compute Delaunay graph using all vertices of an element; (b) instead of using Delaunay edges directly, find the shortest distance between elements represented by each edge; (c) ignore any edges between vertices of the same element; and (d) when calculating subgraphs, add edges among all vertices of an element. See Section 4.2 for algorithm steps.

inter-element edges with two nearby elements will incorrectly fall into two different subgraphs.

We present the multiscale design characteristics recognizer algorithm, AMOEBA [11] extended for spatial regions. The extensions are highlighted in color.

```
procedure MULTISCALE_DESIGN_RECOGNIZER(GraphElements)
 begin
 // (a): Add each vertex of each element
 Graph = CreateDelaunayGraph(GraphElements.Vertices);
 WriteCluster(Graph); // Write the Graph as a cluster

 // (b): Find shortest distance between elements
 //      represented by each edge in the graph
 // In the process, add edges connecting vertices of the
 //      same element to IntraElementEdges
 DistanceMap = FindShortestDistance(Graph,
      IntraElementEdges);

 // (c): Delete edges among vertices of the same element
 Graph.del_edges(IntraElementEdges);
 // Use the shortest distance when computing global mean and
 //      standard deviation
 NumberOfEdges = CalculateMeanandStDev(Graph, GlobalMean,
      GlobalStDev, DistanceMap)

 if (NumberOfEdges <= 1) return;

 for each node v in Graph do {
   EdgeList = Graph.adjacent_edges(v); // Extract edges
        incident to node v
   // Use the shortest distance when computing local mean
   LocalMean = CalculateLocalMean(EdgeList, DistanceMap);
   for each edge e in EdgeList do {
     ToleranceValue = ALPHA * GlobalStDev * GlobalMean /
          LocalMean;
     if (e.distance() >= (GlobalMean + ToleranceValue))
       DeleteEdgeList.append(e);
   }
 }
 Graph.del_edges(DeleteEdgeList); // Eliminate passive
      edges
 if (Graph.degree(v) == 0)
   Graph.delete_node(v);
```

```
 AddIntraElementEdges(Graph); // (d): Add edges among all
      vertices of the same element
 ConnectedComponents(Graph, ComponentNumber); // Calculate
      connected components
 for each connected component c do {
   SubGraph = ConstructSubGraph(Graph, ComponentNumber, c);
   if (NumberOfEdges != SubGraph.number_of_nodes());
     MULTISCALE_DESIGN_RECOGNIZER(SubGraphElements);
 }
end
```

4.3 Recognizer Validation Methodology

Our methodology to validate the multiscale design characteristics recognizer algorithm includes: (1) creating a labeled dataset of scales and clusters present within each of a set of FFWC documents; and (2) deriving performance measures.

4.3.1 Labeled Dataset: Scales and Clusters. Prior work evaluates clustering performance against a set of 30-60 labeled clusters. This includes both quantitative [1, 17] and qualitative [11, 24, 34] evaluation. We aimed to label a comparable dataset, using scales and clusters. We selected FFWC documents from a variety of courses, in proportion to the number of design works produced in each course. At an operational level, the selection criteria were: FFWC documents with a non-trivial number of elements (~30 or more), which were organized at multiple scales (2 or more), i.e., zoom levels. We also selected a few FFWC documents having a single scale or a small number of elements, to validate that the algorithm does not fail in such cases. According to these scales and clusters criteria, from the entire dataset of 235 curations from the studied courses (See Section 4.1), we selected 39 to label.

We developed guidelines for human raters to label clusters across scales. Following the guidelines, two raters labeled scales and clusters within the selected 39 FFWC documents. Through the labeling process, we obtained an average total of 107 scales and 653 clusters. The inter-rater reliability score (Cohen's kappa) for scales was 0.88, indicating a near-perfect [39] agreement. The inter-rater reliability score for clusters was 0.71, indicating a substantial agreement.

A priori, there is no way to ensure that a sample set is representative of a population. Hence, we performed posterior analysis of
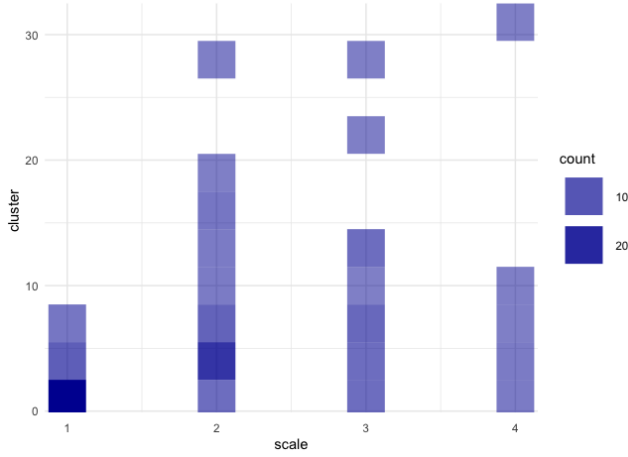
Figure 4: The distribution of the numbers of clusters across scales, in our labeled set, indicates a variety of visual designs, used as the basis for evaluating the multiscale design recognition approach. We used cluster bins of size 3 to smooth the y-axis representation.

the scale and cluster distribution within the labeled set. The distribution plot (Figure 4) indicates a variety of organizations within the labeled set. A majority of the designs use 2 or 3 scales (average = 2.79), which is consistent with prior work (average = 3.10) [36].

4.3.2 Algorithm Performance Measures. Precision, recall, and F-score are commonly used machine learning measures [3]. They have been applied to evaluate algorithm clustering performance in various domains, e.g., interactive document exploration [56], community detection [23], and photo tagging [57]. In spatial domains, these measures have been used for evaluating clusters at a single scale [17]. To evaluate our extended AMOEBA multiscale spatial clustering algorithm, we computed the measures by building on prior multilevel document clustering techniques [56].

Let human labeled clusters be represented by classes (L_1, L_2, L_3, ..., L_c). Next, given a class L_r of size n_r and an algorithm-identified cluster C_i of size n_i , if n_{ri} elements in the cluster C_i belong to L_r, then we compute precision (P), recall (R), and F-Score (F) as:

$$P(L_r, C_i) = \frac{n_{ri}}{n_i} \quad , \quad R(L_r, C_i) = \frac{n_{ri}}{n_r}$$

$$F(L_r, C_i) = \frac{2 * P(L_r, C_i) * R(L_r, C_i)}{P(L_r, C_i) + R(L_r, C_i)}$$

Then, for each L_r, its F-score is the maximum F-score value attained at any node in the tree T of nested clusters, i.e.,

$$F(L_r) = \max_{C_i \in T} F(L_r, C_i)$$

The net F-score of the clustering algorithm is the aggregate individual class F-Score, weighted by its class size:

$$F\text{-}Score_{cluster} = \sum_{r=1}^{c} \frac{n_r}{n} F(L_r)$$

Extending this approach to evaluate scale recognition performance, given s_h human-labeled scales and s_a algorithm-identified

Table 2: Precision, Recall, and F-Score measures, for the two values of α at which we observed peak aggregate performance for scale and cluster recognition in the hyperparameter tuning phase.

	α	Precision	Recall	F-Score
Scale	-0.040	0.531	0.686	0.599
	0.475	0.720	0.628	0.671
Cluster	-0.040	0.658	0.911	0.694
	0.475	0.536	0.979	0.61

scales, for each human-labeled scale M_r, form its pair with the maximum matching scale S_i, based on the sum of $F(L_{ri})$ values, i.e., the F-scores of matching clusters present on the scale i:

$$F(M_r, S_i) = \sum_{ri} \frac{n_r}{n} F(L_{ri}) \qquad F(L_{ri}) : F(L_r) scale\_i\_matches$$

$$F(M_r) = \max_{1 \leq i \leq s_h} F(M_r, S_i)$$

If more than one human-labeled scales match the same algorithm-identified scale, form the pair with the human-labeled scale for which the sum of $F(M_r)$ values is higher. Then, for s_{ha} matches between s_h human-labeled scales and s_a algorithm-identified scales:

$$P_{scale} = \frac{s_{ha}}{s_a} \quad , \quad R_{scale} = \frac{s_{ha}}{s_h}$$

$$F\text{-}Score_{scale} = \frac{2 * P_{scale} * R_{scale}}{P_{scale} + R_{scale}}$$

4.4 Recognizer Validation Results

We measure algorithm performance. We follow with cross-validation.

4.4.1 Precision, Recall, F-Score Performance Measures. Using the approach of Section 4.3.2, we computed precision, recall, and F-Score. In Section 4.2, we described that AMOEBA uses a hyperparameter α to control the heterogeneity of distances among elements that are clustered together. As we adapted AMOEBA in a new domain (i.e., design), we used 10% of the labeled data to tune the hyperparameter.

Analyzing the data, we used F-Score values for tuning, combining precision and recall. Both are important here. Low precision means recognizing extra scales or clusters that human raters did not label. Low recall means not recognizing the human-labeled scales or clusters. We searched both sides of the default hyperparameter value (α=1), following guidance from prior work [11]. We discovered two peaks for aggregate F-Score of scale and cluster recognition, at α values -0.04 and 0.475. We report the validation measures for scale and cluster recognition, for both hyperparameter values (Table 2). We discuss the results in Section 5.

4.4.2 Qualitative Analysis through Visual Inspection. Like prior work, we performed a visual inspection of output clusters, to qualitatively evaluate algorithmic performance. The algorithm correctly recognized scales and clusters in many cases (see an example in Figure 5). At the same time, the algorithm's recognition failed for a few edge cases (see examples in Figure 6). In Section 5.2, we discuss failure cases and consider how to address them.
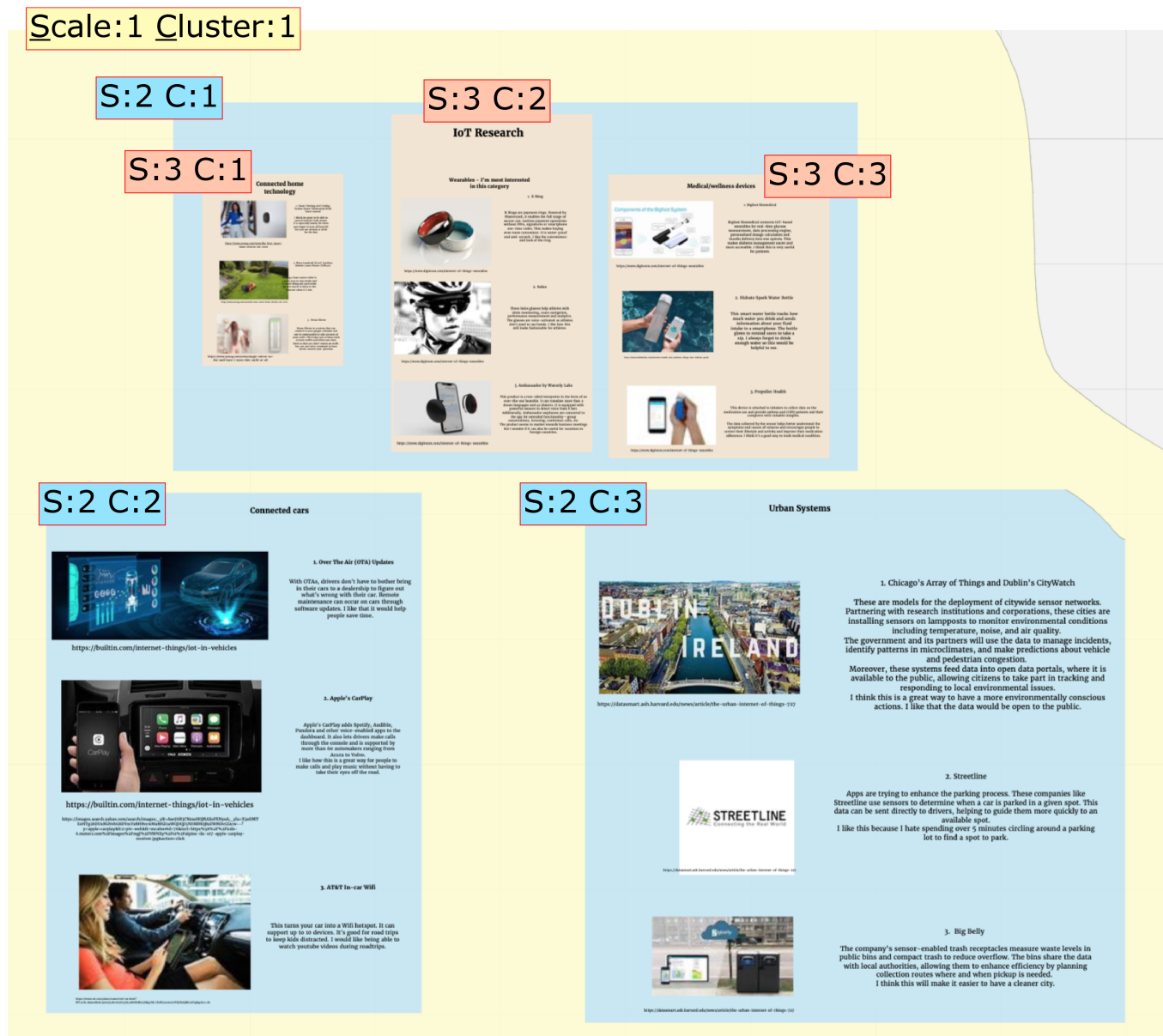
Figure 5: Scale and Cluster Recognition Example. The figure shows the nested scales recognized by the algorithm, with all clusters at a particular scale rendered in the same background color. The outermost scale comprises one cluster—including all design elements—which is rendered in yellow color. The next inner scale has three clusters (one at top and two at bottom), which are rendered in blue. The innermost scale has three clusters—within the top blue cluster—which are rendered in brown.

5 DISCUSSION + IMPLICATIONS

We identify automatic recognition of design characteristics, rooted in practice, as an important area of *human-centered AI*. We build on Shneiderman's advocacy for AI that advances "fundamental human aspirations", such as expressing one's creative potentials, forming social connections, and promoting equity [50]. We see creativity, writ large, and design, as a nexus of approaches and methods, techniques and pedagogy, as fields ripe for developing AI recognizers. The present research contributes to AI supporting human aspirations by advancing the automatic recognition, and so potentially, the practice and learning, of how people represent and communicate abstract ideas.

More specifically, we discuss how this research advances computational recognition of creative visual design, an activity embodying human aspirations, and its implications for supporting design education and people's performance of design work. We follow by focusing on multiscale design, with implications for better recognition of its specific characteristics.

5.1 Recognizing Creative Visual Design

To contextualize with theory, we digress to nomenclature about human cognition. *Convergent thinking* refers to cognitive processes with objective right and wrong answers [12, 26]; *divergent thinking*, in contrast, refers to how people think when performing open ended tasks, with many possible good answers [5, 19]. Machine learning has extensively been used to support convergent thinking tasks. For example, in face recognition, there is one objective right answer that a machine learning model needs to predict [21]. In language modeling, likewise, there is one right answer for the recognition of entities—e.g., person names, organizations, locations—in a given text [41]. Technologies designed to serve ads to people occupy a middle ground [37]. They involve a divergent thinking process of the user. Yet, they can be evaluated with obvious objective characteristics, such as click-through and product purchasing.

In comparison, creative design involves open-ended, divergent thinking tasks. Using AI to support divergent thinking in education contexts exemplifies "rich experiences aimed at advancing human intelligence" [48]. Our research extends a growing chorus of researchers who have used AI and crowds to recognize characteristics of creative design, in contexts spanning idea generation, re-design, and web design. Kerne et al. develops computational recognizers for conceptual aspects of creativity in design, aka *ideation metrics*, such as Fluency (the number of ideas), Flexibility (the variety of ideas), and Novelty (the uniqueness of ideas) [26]. They address visual design, as a holistic ideation metric, without componentizing it, by identifying characterstics such as those of multiscale design, as in the present research. Kumar et al. extract features—e.g., width, height, aspect ratio, and font size—from creative web design and use them to find examples for supporting designers in retargeting their design [31]. Reinecke et al. predict website aesthetics by developing a regression model based on attributes such as color, symmetry, and the number of images and text groups [47]. Gu et al. determined effective design ideas—from a large candidate set for a hierarchical material structure context—by augmenting a convolutional neural network with a self-supervised learning algorithm [18]. Kittur et al. use AI to augment crowd work in searching and filtering candidate ideas from the web to support designers' analogical thinking [27]. Krause et al. crowdsource labeled examples of feedback on student designs, and then use a natural language model to provide suggestions for improving crowd feedback on new designs [29]. Oulasvirta et al.'s Aalto Interface Metrics web service analyzes graphical user interface design and provides metrics for characteristics, such as visual clutter, colorfulness, and whitespace [45].

Our work is complementary to Koch and Oulasvirta's recognizer for hierarchical layout in web page flatlands, using gestalt principles [28]. To our knowledge, our work is the first attempt to computationally identify hierarchical groupings, nested across scales, and accessed via zoomable interfaces. These visual hierarchies, in multiscale design, are often conceptually motivated. Further research in recognizing creative visual design can beneficially connect the conceptual semantics of content elements, along with gestalt principles, and multiscale, visual assemblage.

Our contribution ups the ante on the level of abstraction and communicative sophistication of creative design recognized by artificial intelligence. We developed a multiscale design recognizer,

to help students see their progress on executing this strategy for increasing the legible dimensionality and density of information and so escape the flatland of document representation. In doing this, we contribute understanding of how computing systems can recognize visual, spatial characteristics of design representations, which facilitate "crossing through scales" [35], which, "is about controlling simultaneously and in the same way, the general and the specific, the close and the far" [6].

Prior contextualized findings further motivate the value of the present AI algorithmic contribution. Multiscale design through FFWC has been found to enable design students to refer to and reuse ideas; it facilitates consistency in the presentation of ideas across project deliverables [4]. Prior research in project-based learning contexts shows how students' engagement in multiscale design through FFWC documents supports them in creating and communicating relationships among a large number of ideas [36].

Implications. Prior research notes diverse potential application areas for recognition of hierarchical groupings in visual design, including automated feedback on design work and computer-generated interfaces [28]. The present research can be used to extend opportunities identified by prior work, enabling people to obtain on-demand feedback for improving how their design utilizes space and scale. Likewise, similar to design mining [30], our work has the potential to support people by suggesting alternative multiscale design layouts exhibiting similar characteristics, i.e., the number of scales and nested clusters across scales.

In design education, specifically, providing students with on demand feedback and example layouts has the potential to scaffold learning. Such scaffolding will be particularly useful for novice designers—like Sue in our scenario, working on a project the night before it is due—who get stuck [44] when working on projects involving multiscale design. Design fixation [22] can thus be overcome through on-demand feedback based on this recognizer. Multiscale designers use tools such as Photoshop and Illustrator in diverse course contexts. They would benefit from incorporation of multiscale design recognizers in these tools.

5.2 Recognizing Multiscale Design

Our contribution leverages identifying multiscale design as a key component of visual design, understanding the roles of multiscale design in graphical communication, and then operationalizing this understanding with computational recognizers. We consider opportunities for improving the current multiscale design recognizers recognizer and for building better ones.

In the current recognizer, we adapted the AMOEBA algorithm for multiscale clustering of design work. We evaluated the performance of the adapted algorithm using precision, recall, and F-score measures (Table 2). As prior work has not quantitatively evaluated multiscale clustering in the spatial domain, no direct comparisons can be made. The performance level, however, is comparable to state-of-the-art multiscale clustering in the non-spatial community detection domain [23]. We also compare the cluster F-Score measure (0.694) with that for single-scale spatial clustering (0.79-1.0) [17], finding a drop of 0.1 to 0.3 points in the performance. This makes sense because determining nested clusters is more complex. Drawing analogy from natural language processing, the state-of-the-art model for named entity recognition achieves 0.95 F-score, entity

(a)

(b)
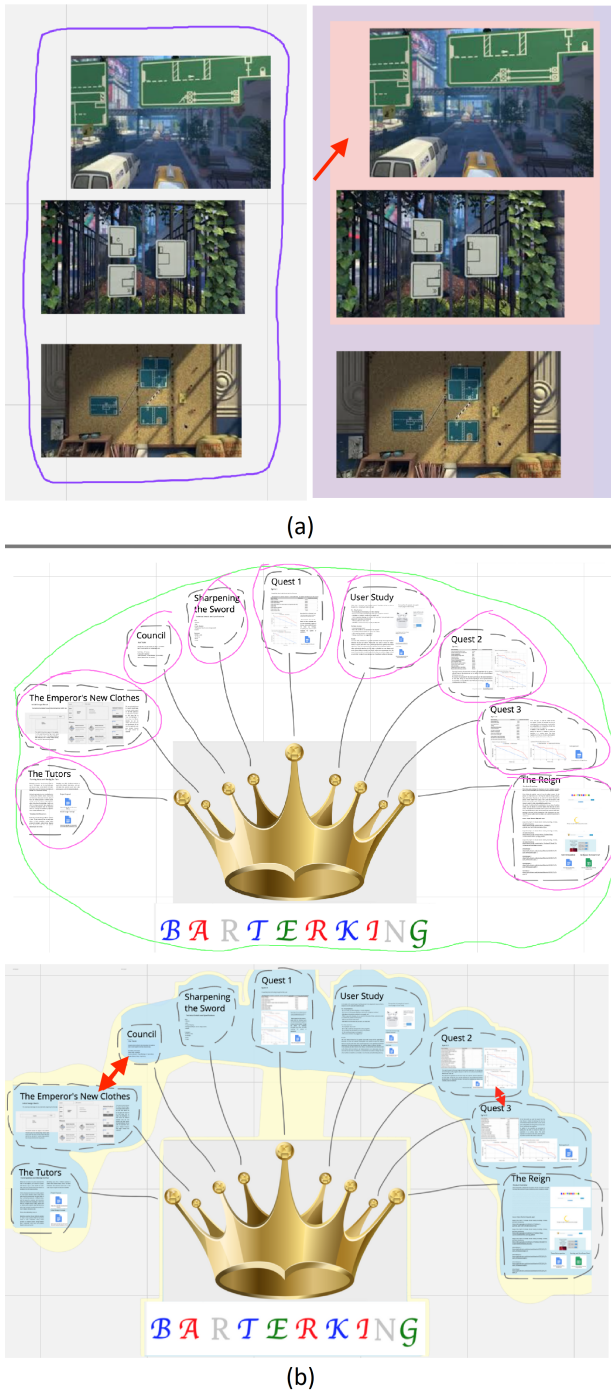
Figure 6: Patterns where current recognition falls short: (a) Image similarity: A human rater labeled (left) a cluster of all images vs algorithm (right) that does not account for the similarity of blocks in bottom images; and (b) Enclosing region: A human rater labeled (top) more clusters vs algorithm (bottom) operating based on spatial distances and not accounting for similarities in the enclosing sketched region.

linking 0.82, and relationship extraction 0.76 [49]. As abstraction in the cognitive task increases, performance decreases.

Implications. Our investigation contributes a baseline algorithm and evaluation to the research community. Through our validation, we identify opportunities for improving recognition performance. Figure 6 (a) shows that a human rater labeled all image elements on one scale. The current algorithm, based on spatial positioning and dimensions, did not account for the similarity of blocks existing within bottom images. Such cases resulted in identifying extra scales and clusters, and thus, lower scale and cluster precision. To address such failures, in the future, we can utilize state-of-the-art, *fine-grained* image similarity approaches [54] and include additional features in the algorithm. In Figure 6 (b), a human rater labeled clusters due to similarities in the enclosing sketched region. Such failures, where the algorithm does not identify some clusters labeled by a human rater and/or identifies a few additional ones, result in lower cluster precision and recall. To address these, we advocate extending the present algorithm with the gestalt principle of 'common region' [53], which Koch and Oulasvirta employ in recognizing hierarchical layout in web pages [28].

Improving the algorithmic performance will open opportunities for recognizing further characteristics of spatial nesting in multiscale design. Spatial nesting is a form of organizing ideas in hierarchies. Drawing from Gentner's structure-mapping framework [15], hierarchies form the basis of mapping relations among different ideas. The development of hierarchies supports two fundamental operations in creative cognition: abstraction and analogy [55]. Using scale and cluster recognition as a basis, new design recognizers can be built for abstraction and analogy characteristics. First, ideas contained within all nested clusters of elements can be extracted through multimedia processing of information—e.g., text, image, and video—stored within respective clippings (Figure 2). Then, these extracted ideas can be compared across scales and clusters by using WordNet [40]—a tree-like semantic representation of words—to recognize, as well as recommend abstractions and analogies. Future research can beneficially investigate the usefulness of these characteristics in diverse creative education—e.g., mechanical engineering, computer science, and architecture—contexts.

6 CONCLUSION

The present research demonstrates an AI technique for supporting "fundamental human aspirations," to creatively re-imagine representations of our world. We present a human-centered AI algorithm, for supporting designers in understanding their progress in using design strategies to increase the legible dimensionality of information on the screen. Our approach extends a prior spatial clustering algorithm to recognize multiscale design characteristics in students' FFWC documents. We contribute baseline performance for multiscale design recognition. We identify opportunities for better performance and widening frontiers for creative design recognition. We observe the potential of this research to contribute toward supporting people with on-demand feedback on their design work.

Supporting creative learning has the potential to stimulate economic growth and innovation [42]. The current results provide evidence to motivate further investigations, e.g., whether and how the recognized characteristics affect educational outcomes in creative learning contexts. Investigations involving diverse machine

learning techniques, for diverse design characteristics, are expected to advance creative design recognition and support valuable contexts of human learning and growth. Likewise, future work will benefit from broadening the space of investigations into diverse document structures and characteristic recognizers to support people's engagement in creative tasks and activities.

7 ACKNOWLEDGEMENT

REFERENCES

[1] Alper Aksaç, Tansel Özyer, and Reda Alhajj. 2019. CutESC: cutting edge spatial clustering technique based on proximity graphs. *Pattern Recognition* 96 (2019), 106948.

[2] Alexander Alberro and Blake Stimson. 1999. *Conceptual art: a critical anthology.* MIT press.

[3] Ethem Alpaydin. 2020. *Introduction to machine learning.* MIT press.

[4] Gabriel Britain, Ajit Jain, Nic Lupfer, Andruid Kerne, Aaron Perrine, Jinsil Seo, and Annie Sungkajun. 2020. Design is (A)live: An Environment Integrating Ideation and Assessment. In *CHI Late-Breaking Work.* ACM, 1–8.

[5] Bill Buxton. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[6] Michel Corajoud. 2004. L'horizon. *Face* (Mars 2004).

[7] Min Deng, Qiliang Liu, Tao Cheng, and Yan Shi. 2011. An adaptive spatial clustering algorithm based on Delaunay triangulation. *Computers, Environment and Urban Systems* 35, 4 (2011), 320–332.

[8] Joel AF dos Santos, Christiano Braga, Débora C Muchaluat-Saade, Cécile Roisin, and Nabil Layaïda. 2015. Spatio-temporal validation of multimedia documents. In *Proc. DocEng.* ACM, 133–142.

[9] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. 2015. The Delaunay document layout descriptor. In *Proc. DocEng.* ACM, 167–175.

[10] Vladimir Estivill-Castro and Ickjai Lee. 2000. Autoclust: Automatic clustering via boundary extraction for mining massive point-data sets. In *Proc. Geocomputation.*

[11] V. Estivill-Castro and Ickjai Lee. 2002. Multi-level clustering and its visualization for exploratory spatial analysis. *GeoInformatica* 6, 2 (2002), 123–152.

[12] R.A. Finke, T.B. Ward, and S.M. Smith. 1992. *Creative Cognition: Theory, Research, and Applications.* MIT Press.

[13] Luoting Fu and Levent Burak Kara. 2011. From engineering diagrams to engineering models: Visual recognition and applications. *Computer-Aided Design* 43, 3 (2011), 278–292.

[14] A. Gabdulkhakova and T. Hassan. 2012. Document understanding of graphical content in natively digital PDF documents. In *Proc. DocEng.* ACM, 137–140.

[15] Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science* 7, 2 (1983), 155–170.

[16] John S Gero and Mary Lou Maher. 1993. *Modeling creativity and knowledge-based creative design.* Psychology Press.

[17] Tony H Grubesic, Ran Wei, and Alan T Murray. 2014. Spatial clustering overview and comparison: Accuracy, sensitivity, and computational expense. *Annals of the Association of American Geographers* 104, 6 (2014), 1134–1156.

[18] Grace X Gu, Chun-Teh Chen, Deon J Richmond, and Markus J Buehler. 2018. Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment. *Materials Horizons* 5, 5 (2018), 939–945.

[19] J.P. Guilford. 1956. The structure of intellect. *Psych Bulletin* 53 (1956), 267–293.

[20] Mohammad Nehal Hasnine, Gökhan Akçapınar, Kousuke Mouri, and Hiroshi Ueda. 2020. An Intelligent Ubiquitous Learning Environment and Analytics on Images for Contextual Factors Analysis. *Applied Sciences* 10, 24 (2020), 8996.

[21] Anil K Jain and Stan Z Li. 2011. *Handbook of face recognition.* Vol. 1. Springer.

[22] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.

[23] Liang Jiang, Leilei Shi, Lu Liu, Jingjing Yao, and Muhammad Ali Yousuf. 2019. User interest community detection on social media using collaborative filtering. *Wireless networks* (2019), 1–7.

[24] George Karypis, Eui-Hong Han, and Vipin Kumar. 1999. Chameleon: Hierarchical clustering using dynamic modeling. *Computer* 32, 8 (1999), 68–75.

[25] Andruid Kerne, Nic Lupfer, Rhema Linder, Yin Qu, Alyssa Valdez, Ajit Jain, and Others. 2017. Strategies of Free-Form Web Curation: Processes of Creative Engagement with Prior Work. In *Proc. C&C.* ACM, 380–392.

[26] Andruid Kerne, Andrew M Webb, Steven M Smith, Rhema Linder, Nic Lupfer, Yin Qu, Jon Moeller, and Sashikanth Damaraju. 2014. Using metrics of curation to evaluate information-based ideation. *ACM TOCHI* 21, 3 (2014), 1–48.

[27] Aniket Kittur, Lixiu Yu, Tom Hope, Joel Chan, Hila Lifshitz-Assaf, Karni Gilon, and Others. 2019. Scaling up analogical innovation with crowds and AI. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1870–1877.

[28] Janin Koch and Antti Oulasvirta. 2016. Computational layout perception using gestalt laws. In *Proc. CHI Extended Abstracts.* 1423–1429.

[29] Markus Krause, Tom Garncarz, JiaoJiao Song, Elizabeth M Gerber, Brian P Bailey, and Steven P Dow. 2017. Critique style guide: Improving crowdsourced design feedback with a natural language model. In *Proc. CHI.* ACM, 4627–4639.

[30] Ranjitha Kumar, Arvind Satyanarayan, Cesar Torres, Maxine Lim, Salman Ahmad, Scott R Klemmer, and Jerry O Talton. 2013. Webzeitgeist: design mining the web. In *Proc. CHI.* ACM, 3083–3092.

[31] Ranjitha Kumar, Jerry Talton, Salman Ahmad, and Scott Klemmer. 2011. Bricolage: example-based retargeting for web design. In *Proc. CHI.* ACM, 2197–2206.

[32] Der-Tsai Lee and Bruce J Schachter. 1980. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences* 9, 3 (1980), 219–242.

[33] Lucy R Lippard. 2007. *Dadas on art: Tzara, Arp, Duchamp and others.* Courier Corporation.

[34] Qiliang Liu, Min Deng, Yan Shi, and Jiaqiu Wang. 2012. A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity. *Computers & Geosciences* 46 (2012), 296–309.

[35] Nic Lupfer, Hannah Fowler, Alyssa Valdez, Andrew Webb, Jeremy Merrill, Galen Newman, and Andruid Kerne. 2018. Multiscale Design Strategies in a Landscape Architecture Classroom. In *Proc. Designing Interactive Systems.* ACM, 1081–1093.

[36] Nic Lupfer, Andruid Kerne, Rhema Linder, Hannah Fowler, and Others. 2019. Multiscale Design Curation: Supporting Computer Science Students' Iterative and Reflective Creative Processes. In *Proc. C&C.* ACM, 233–245.

[37] Thomas J Madden, Chris T Allen, and Jacquelyn L Twible. 1988. Attitude toward the ad: An assessment of diverse measurement indices under different processing "sets". *Journal of marketing research* 25, 3 (1988), 242–252.

[38] Catherine C Marshall. 1998. Toward an ecology of hypertext annotation. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia.* 40–49.

[39] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica* 22, 3 (2012), 276–282.

[40] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[41] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.

[42] National Academy of Engineering. 2010. *Rising Above the Gathering Storm, Revisited: Rapidly Approaching Category 5.* The National Academies Press.

[43] Yeonjoo Oh, Suguru Ishizaki, Mark D Gross, and Ellen Yi-Luen Do. 2013. A theoretical framework of design critiquing in architecture studios. *Design Studies* 34, 3 (2013), 302–325.

[44] Jane Osmond and Michael Tovey. 2015. The Threshold of Uncertainty in Teaching Design. *Design and Technology Education* 20, 2 (2015), 50–57.

[45] Antti Oulasvirta, Samuli De Pascale, Janin Koch, and Others. 2018. Aalto Interface Metrics (AIM) A Service and Codebase for Computational GUI Evaluation. In *UIST Adjunct Proceedings.* 16–19.

[46] Yin Qu, Andruid Kerne, Nic Lupfer, Rhema Linder, and Ajit Jain. 2014. Metadata type system: Integrate presentation, data models and extraction to enable exploratory browsing interfaces. In *Proc. EICS.* ACM, 107–116.

[47] Katharina Reinecke, Tom Yeh, Luke Miratrix, and Others. 2013. Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In *Proc. CHI.* ACM, 2049–2058.

[48] Thanassis Rikakis, Aisling Kelliher, Jia-Bin Huang, and Hari Sundaram. 2018. Progressive cyber-human intelligence for social good. *Interactions* 25, 4 (2018), 52–56.

[49] S Ruder. 2020. NLP-progress. *London (UK): Sebastian Ruder (accessed 2020-09-01). https://nlpprogress. com* (2020).

[50] Ben Shneiderman. 2021. Human-Centered AI. *Issues in Science and Technology* 37, 2 (2021), 56–61.

[51] Pao Siangliulue, Joel Chan, Steven P Dow, and Krzysztof Z Gajos. 2016. IdeaHound: improving large-scale collaborative ideation with crowd-powered real-time semantic modeling. In *Proc. UIST.* ACM, 609–624.

[52] Edward R Tufte, Nora Hillman Goeler, and Richard Benson. 1990. *Envisioning information.* Vol. 126. Graphics press Cheshire, CT.

[53] Johan Wagemans, James H Elder, Michael Kubovy, Stephen E Palmer, and Others. 2012. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. *Psychological bulletin* 138, 6 (2012), 1172.

[54] Jiang Wang, Yang Song, Thomas Leung, and Others. 2014. Learning fine-grained image similarity with deep ranking. In *Proc. CVPR.* IEEE, 1386–1393.

[55] Hans Welling. 2007. Four mental operations in creative cognition: The importance of abstraction. *Creativity Research Journal* 19, 2-3 (2007), 163–177.

[56] Ying Zhao and George Karypis. 2002. Evaluation of hierarchical clustering algorithms for document datasets. In *Proc. CIKM.* 515–524.

[57] Chunhui Zhu, Fang Wen, and Jian Sun. 2011. A rank-order distance based clustering algorithm for face tagging. In *Proc. CVPR.* IEEE, 481–488.