d.tour: Style-based Exploration of Design Example Galleries

Daniel Ritchie, Ankita Arvind Kejriwal, Scott R. Klemmer
Stanford University HCI Group
Department of Computer Science
Stanford, CA, USA
(dritchie, ankitak, srk)@cs.stanford.edu

ABSTRACT

In design, people often seek examples for inspiration. However, current example-finding practices suffer many drawbacks: templates present designs without a usage context; search engines can only examine the text on a page. This paper introduces exploratory techniques for finding relevant and inspiring design examples. These novel techniques include searching by stylistic similarity to a known example design and searching by stylistic keyword. These interactions are manifest in d.tour, a style-based design exploration tool. d.tour presents a curated database of Web pages as an explorable design gallery. It extracts and analyzes design features of these pages, allowing it to process style-based queries and recommend designs to the user. d.tour's gallery interface decreases the gulfs of execution and evaluation for design example-finding.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Design, examples, search

INTRODUCTION

People often begin design work by referring to, copying, or adapting existing designs [19, 12]. Prior work has provided techniques to adapt the content [13] and the style [18] of example pages to designs.

However, finding relevant design examples remains a challenge. People commonly query search engines for pages with similar content to their own. However, search engines demand keywords, which can be hard to articulate [12]. This is especially true for design novices, as they often engage in exploratory search without clear stylistic goals. Furthermore, current search engines offer little help because they index and present only the *contents* of a page, not its *design style*. Another common approach is to browse design template repositories. But templates present designs without showing them in use, so it can be hard to gauge their value [19].

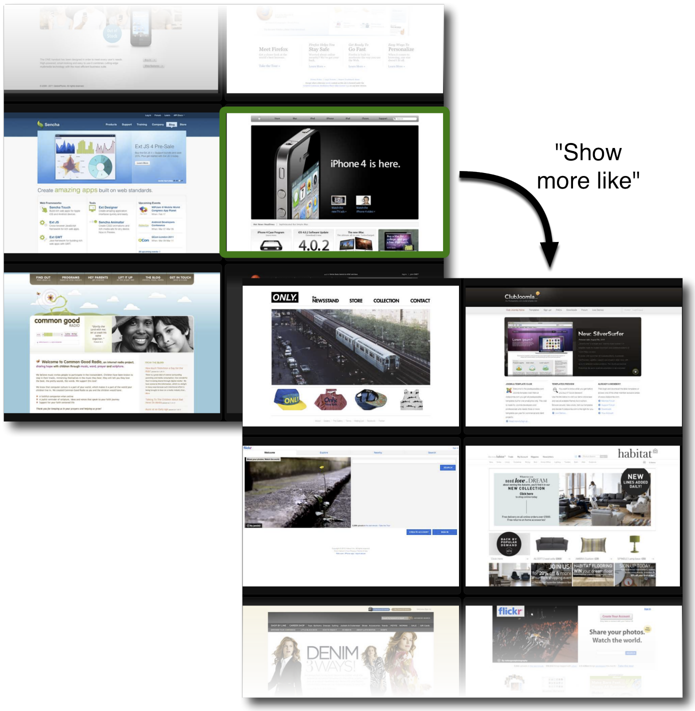


Figure 1: d.tour enables search for designs by style. The “Show more like" interaction supports search by stylistic similarity, using examples as queries (design outlined in green). The results (second set of designs) incorporate features from these examples.

How can tools enable people to explore designs for inspiration? Prior work has introduced two main strategies for design search. One strategy is to retrieve designs given a rough layout sketch [10]. However, this requires people to have concrete layout ideas before they can begin exploring examples. They also cannot explore along non-layout design dimensions. The second strategy is a browsable design gallery [19, 21]. Existing gallery strategies have been limited to browsing. We employ a gallery approach, introducing techniques for search, recommendation, filtering and style-based exploration.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
UIST'11, October 16-19, 2011, Santa Barbara, CA, USA.
Copyright 2011 ACM 978-1-4503-0716-1/11/10...$10.00.

This paper introduces techniques for finding design examples by style, which are manifest in d.tour—an exploratory search tool for Web design. In many domains, it is difficult to specify what one wants, but easy to provide positive and negative examples. Recommender systems leverage this insight [3], and d.tour extends it to the design domain. It can be easier to specify design goals by visual example than with language, so d.tour allows people to search its gallery for designs that are stylistically similar/dissimilar to an example design. People can also search by high-level style terms such as ‘minimal’ and ‘clean.’ To support such queries, d.tour automatically extracts style-related features from a curated database of designs. The d.tour project investigates example-finding techniques for visual design; the prototype presented in this paper focuses on Web design.

d.tour contributes the following:

- An interface and interaction techniques for style-based gallery exploration.
- A curated database of Web pages for people to search.
- Methods for extracting stylistic and structural features from Web pages; these features make our novel search modalities possible.
- A recommender system that processes queries and returns relevant pages from the database.

The d.tour Web interface is located at http://bricolage.stanford .edu/search.

We believe that style-based exploration of a design gallery can help people of all skill levels find relevant and inspiring design examples more effectively. To get an initial sense of the utility of this direction, an online study asked participants to find inspirational design examples using either the d.tour prototype or keyword Web search. Participants who used d.tour reported a positive example-finding experience compared to those who used Web search. The current d.tour prototype uses a 300-page corpus of designs selected from design blogs, award lists, and the Alexa Top 100. In future work, it will be important to investigate the effect of corpus size on user experience.

The remainder of this paper is organized as follows: first, we describe the d.tour interface and the search interactions it supports. We then identify a set of design features and describe algorithms to extract them from Web pages and use them to answer search queries. We describe the online study and our experimental results, and conclude with discussion and directions for future work.

USER INTERFACE

The d.tour interface comprises five components (Figure 2):

(a) *Search Results*: Presents search results as interactive thumbnails.

(b) *Search box*: Supports query by text, color, and style. Allows users to add their own pages via URL.

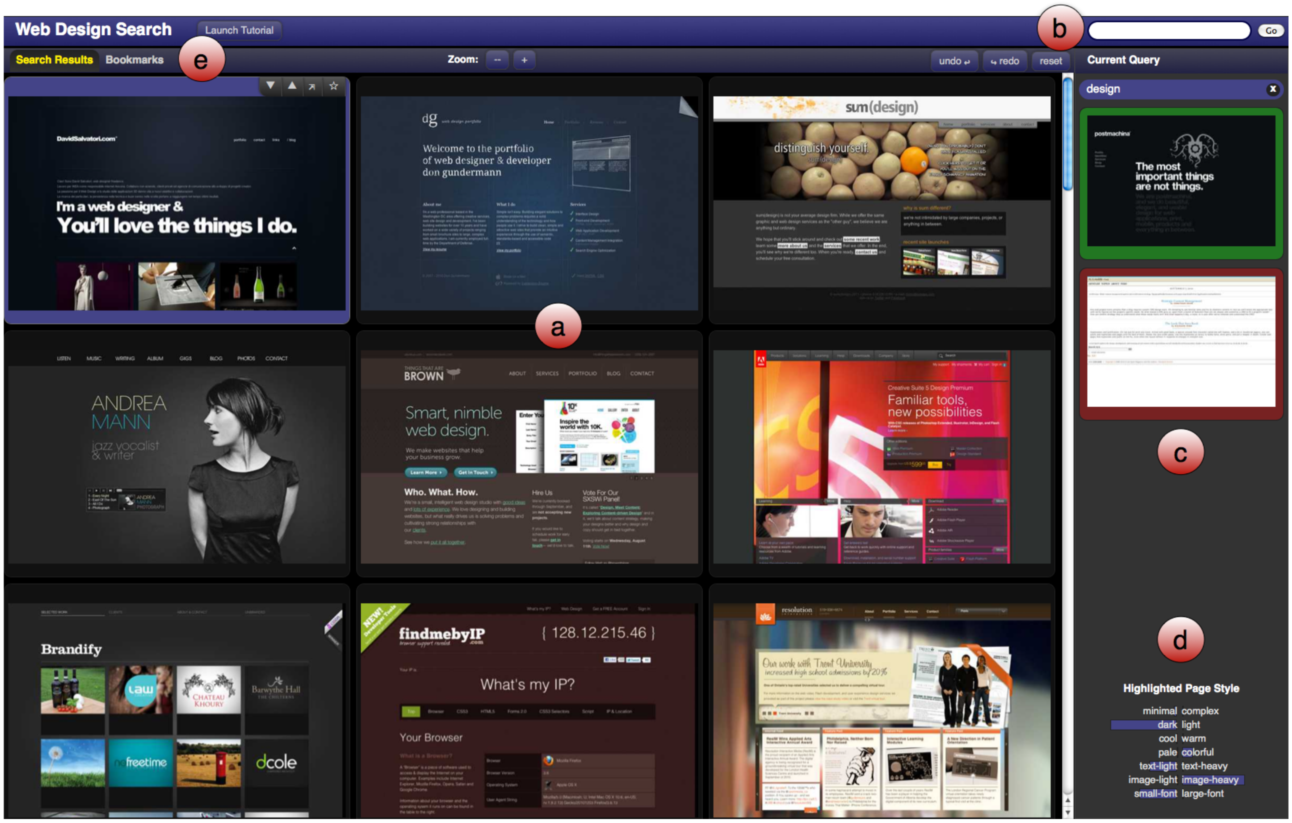


Figure 2: The d.tour Web interface: (a) Search Results, (b) Search box, (c) Current Query, (d) Ambient Style Display, and (e) Bookmarks. In (c), the user has searched for ‘design’ and has given a positive rating to one page and a negative rating to another. The Search Results (a) displays pages matching this query.

(c) *Current Query*: Displays all keywords and Web pages currently influencing the search results.
(d) *Ambient Style Display*: Helps people understand why pages are recommended.
(e) *Bookmarks*: Keeps track of interesting pages.

d.tour initially presents a random sample of pages from the system's database. The user can search by querying for designs similar/dissimilar to design examples or by entering keywords into the search box.

Search Results

d.tour enables people to use design examples as a query interface by providing 'Show more like' (▲) and 'Show less like' (▼) controls that update the design gallery accordingly (Figure 1). These controls become visible when hovering over a page, as seen in the first result in Figure 2a. Other interactions include clicking a page thumbnail to expand the page to full size, ↗ to open it in a new tab, and ☆ to bookmark and add notes to the page.

Search box

With d.tour's search box, people can search with keyword queries and URLs. First, the user can search for documents that contain a particular *text* string. The user can also search by *color* keywords, *style* terms, or any combination of the three. For example, searching for 'colorful image-heavy' retrieves colorful pages that make heavy use of images (Figure 3). The d.tour prototype supports style keywords that arose frequently in our discussions with potential users; this set could of course be expanded. The user can query for designs similar to any page on the Web by entering its URL into the search box.

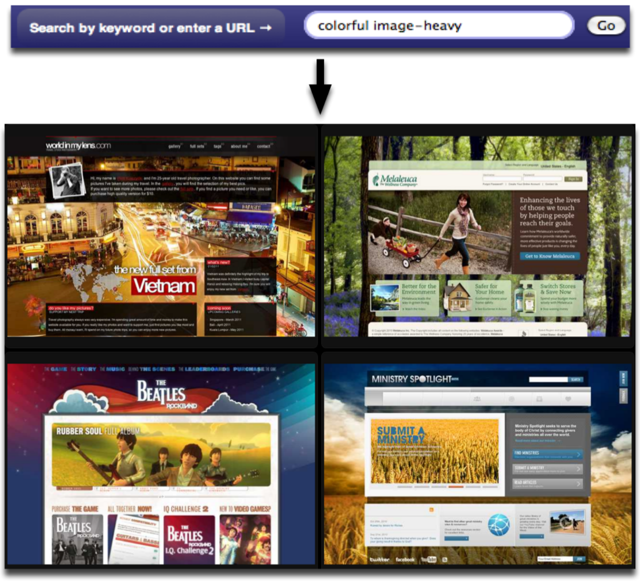


Figure 3: Performing style keyword search with d.tour. Searching for 'colorful image-heavy' retrieves colorful pages that make heavy use of images.

Current Query

This right-hand pane displays the keyword and page queries that specify the current state of the design gallery. The user can freely modify this state by changing or removing items.

Pilot studies indicated that people often like only *some* attributes of design examples; example-based querying could be more efficient if users provided information about their rationale for liking a page. Building simple interfaces for users to convey rationale is an important open issue in recommender systems [30]. After experimenting with several alternatives, we found that checkboxes for *Color* and *Layout* work well.

Ambient Style Display

Explaining recommendations can improve user experience, build trust, and support evaluation of system conclusions [28, 25, 9]. d.tour provides the Ambient Style Display to help explain search results. This bar chart visualizes the aggregate stylistic attributes of the current example queries. When a user hovers over a page, the display provides instant feedback by showing stylistic attributes for that particular page (Figures 2d and 4).

The Ambient Style Display also makes the supported style search terms transparent to the user. Instead of typing style keywords into the search box, the user can click on any style term displayed here to search for pages that are strong in that style.

Bookmarks

The 'Bookmarks' tab allows people to review the pages they have bookmarked, edit their notes, and delete bookmarks. Bookmarks persist when a user resets the Current Query; this allows the user to maintain a collection of interesting pages across several search sessions.

IMPLEMENTATION

This section describes the current prototype system behind the d.tour interface.

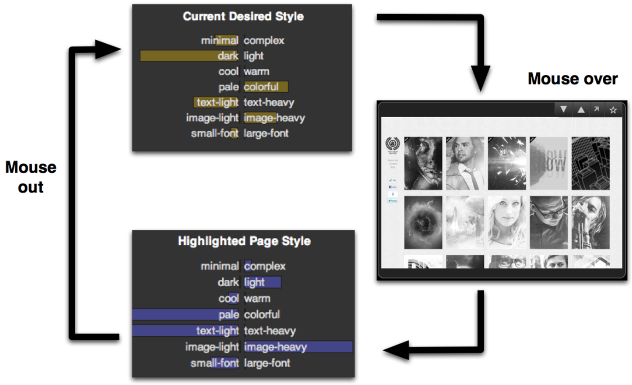


Figure 4: The Ambient Style Display shows the aggregate style attributes of the current example queries. When the user mouses over a page thumbnail, it instead shows that page's stylistic attributes.

Database

The current prototype includes around 300 Web pages listed on sites such as webbyawards.com, designmeltdown.com, ilovetypography.com, sixrevisions.com, and the Alexa Top 100. The complete contents of the database can be found at bricolage.stanford.edu/database. As d.tour automatically extracts features, the database could easily expand by crawling the Web. Techniques such as Website rating prediction [15] could help automatically pick out good designs. People can already add pages to d.tour's database with the search box; given enough users, the database could expand by crowdsourcing. Social moderation techniques [20] could help maintain quality as the corpus grows organically.

System Architecture

All utilities for processing pages and extracting features are implemented in C++ with the Qt framework; we use the QtWebKit API to interface with the WebKit browser engine.

The d.tour search interface is written in HTML, JavaScript, and CSS; it supports HTML5-compliant browsers such as Firefox, Safari, and Chrome. The Web server is also written in JavaScript using node.js [1].

FEATURE EXTRACTION

To support interactive search, d.tour precomputes style features for every page in its database and any new page added during runtime. d.tour focuses on global features that describe a design's use of space, color, text, and images (Figure 5). Our choice of these features was informed by interviews with designers and principles from the design literature [27]. From our observations, these features compose to predict higher-level stylistic attributes. For example, a shallow DOM tree, large foreground-to-background ratio, and few colors form a good predictor of a 'minimalist' style. Many of these features are derived from the page's Document Object Model (DOM) tree. Features such as "number of fonts" can be directly read from the DOM tree. Other features require segmentation. d.tour applies the Bento page segmentation algorithm [18] to ensure that the DOM tree accurately reflects the page's visual layout. The system also computes a color histogram of the rendered page, quantizing the image into 216 colors that evenly sample the RGB color cube. After extraction, the system standardizes all features, so that those with typically large values (such as the size of image elements) do not unduly dominate the others.

Use of Space	DOM tree depth (min, max, mean, stddev), # of DOM leaf nodes, document width/height, amount of separation between content blocks, foreground/background ratio, overlapping element area, connected components in rendered image (#, minarea, maxarea, meanarea)
Use of Color	Color (mean, stddev), saturation (min, max, mean, stddev), value (min, max, mean, stddev), # of colors (in DOM, in rendered page), most dominant color, most dominant text color, # of dominant colors, text-to-background contrast (min, max, mean, stddev), histogram
Use of Text	# of words in page, # of words per block (min, max, mean, stddev), # of fonts, font size (min, max, mode, mean)
Use of Images	# of images, aspectRatio (min, max, mean, stddev), area (min, max, mean, stddev), complexity (min, max, mean, stddev)

Figure 5: d.tour leverages these global page features, which were selected based on discussions with designers and principles from the design literature.

d.tour computes additional features for the local structure of Web pages. The Bricolage algorithm [18] transfers the contents of a source page into the style of a target page by computing a mapping between the two pages' DOM trees. It uses an optimization framework, so each mapping has an associated cost. Mapping between two design-similar pages induces a low cost, whereas mapping between dissimilar ones induces high cost. This cost can be viewed as a 'stylistic distance' between Web pages, and d.tour exploits this by precomputing mapping costs between all pairs of pages in the database. The Bricolage mapping costs are computed offline.

All the features for each page are concatenated into one combined vector. This is d.tour's final representation of a page.

We could have treated Web pages purely as images and used content-based image retrieval (CBIR) to answer queries [24, 7, 4]. However, the low-level pixel features used in CBIR were developed with natural images in mind; when applied to Web pages, they capture little information about design style. Recent work in GUI automation and reverse engineering computes patch-based image features to identify interface components [32, 5]. While this strategy works well for exactly matching individual components in designs, local patches do not capture the gestalt style and structure of designs themselves.

QUERY PROCESSING

d.tour uses the text contents of its pages and the combined feature vectors described above to respond to different types of queries.

'Show More/Less Like'

To represent the user's overall preferences according to the current query, the system takes a weighted average q of all feature vectors for pages rated via 'show more/less like.' Let M be the set of feature vectors for all 'show more like' pages and L be the set of vectors for all 'show less like' pages. Then:

$$q = \frac{1}{(|M| + |L|)}(\sum_{\mathbf{m} \in M} \mathbf{m} - \sum_{\mathbf{l} \in L} \mathbf{l})$$

If the user has turned off the *Color* toggle for a particular page, all color-related features are set to zero in that page's feature vector. If the *Layout* toggle is off, all other features are set to zero. The system then ranks each page in the database by its feature vector's cosine distance to the query vector q and returns the ranked list.

Text Keyword Search

For each page in the database, the system computes a TF-IDF score for each word in the page's document text [26]. It returns a ranked list of the pages whose combined scores for the query terms are greater than zero.

Color Keyword Search

The system converts each color keyword into an RGB color c^* for that word using a hard-coded lookup table; the current prototype supports twelve common color words. To determine how well a page matches the query color, it computes a score s based on the page's histogram h. h is a set of pairs (c_i, f_i), where c_i is an RGB color and f_i is the fraction of pixels in the page whose nearest histogram color is c_i . Then:

$$s = \sum_{i=1}^{|h|} s_i$$

where, for each histogram pair (c_i, f_i)

$$d_i = ||c^* - c_i||_2 \quad \text{and} \quad s_i = \begin{cases} \frac{f_i}{1+d_i^2} & \text{if } d_i < \epsilon, \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, frequently-occurring colors that are close to the query color should have a high score contribution s_i. We clamp the contribution to zero if the color is not sufficiently close; this prevents extremely frequent but only marginally similar colors from skewing the score. We use a value of $\frac{\sqrt{3}}{3}$—one-third the diagonal length of the RGB color cube—for the threshold ϵ, which works well in practice. The system returns as search results only the pages with s greater than a heuristically determined threshold.

Style Keyword Search

We informally asked participants to tag Web pages with stylistic descriptor words. Some words occurred much more frequently than others. Our next step was to operationalize these keywords as compositions of style features. For instance, *minimal* pages have few DOM leaf nodes and few image connected components. Style keyword search works by consulting a rule for each query term, scoring pages that closely conform to the rule higher than those that do not. The system multiplies a page's scores for each query term and heuristically thresholds the search results. The current prototype supports 35 terms—including synonyms—that map to 14 unique rules (Figure 6). Terms corresponding to these unique rules also drive the Ambient Style Display interface component described in the User Interface section.

Minimal, Minimalistic, Clean, Simple, Plain, Sparse Few DOM leaves; few image connected components	↔	**Complex, Complicated, Busy, Dense, Intricate** Many DOM leaves; many image connected components
Dark Low mean color value	↔	**Light, Bright** High mean color value
Cool, Cold Mean color close to pale blue	↔	**Warm** Mean color close to orange
Pale, Monochrome, Faded Low mean saturation	↔	**Colorful, Saturated, Vivid** High mean saturation; high number of colors
Text-light, Text-sparse Few words on the page	↔	**Text-heavy, Text-centric, Text-focused** Many words on the page
Image-light, Image-sparse Low total image area	↔	**Image-heavy, Image-centric, Image-focused** High total image area
Small-font Small mean font size; small mode font size	↔	**Large-font** Large mean font size; large mode font size

Figure 6: The current d.tour prototype invokes these rules when the user searches for the bold terms. All the features in any single rule are weighted equally.

Supporting a larger vocabulary would almost certainly improve the search experience. However, many style terms have complex, subjective meanings and resist d.tour's current heuristic strategy. For instance, it is not clear how 'professional' or 'fun' map onto our feature set. Employing crowdsourcing and machine learning to expand the keyword set is an important direction for future work.

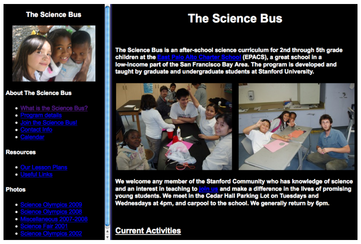
EVALUATION

Method

An online study instructed participants to find a set of inspirational design examples—*Inspiration Boards*—for redesigning a Website. Participants were randomly assigned to use either d.tour or a Web search engine to find examples. Afterwards, they provided feedback about their search strategies and experience.

Participants We recruited 40 participants: 20 through student email lists at our university and 20 more via Amazon Mechanical Turk. Participants answered two questions about their prior Web design experience: "How many Web pages have you designed?" and "Of these, how many were paid, professional assignments?" Participants of different expertise levels were distributed across using either d.tour or a Web search engine by random assignment.

Study Design All participants searched for inspirational design examples for the same client: The Science Bus, an after-school science education program [2]. We chose this group because we believed that most participants could sympathize with its mission, care about the study task, and fully apply themselves [11]. Also, the Website to be redesigned was not heavily styled; we believed that such a raw starting point would help participants avoid design fixation [16].



The Science Bus

The Science Bus is an after-school science curriculum for 2nd through 5th grade children at the East Palo Alto Charter School (EPACS), a great school in a low-income part of the San Francisco Bay Area. The program is developed and taught by graduate and undergraduate students at Stanford University.

We welcome any member of the Stanford Community who has knowledge of science and an interest in teaching to join us and make a difference in the lives of promising young students. We meet in the Cedar Hall Parking Lot on Tuesdays and Wednesdays at 4pm, and carpool to the school. We generally return by 6pm.

Current Activities

The online system randomly assigned each participant to use either d.tour or a Web search engine to search for examples. Each participant submitted URLs for four inspirational design examples and provided reasons for choosing each page. Participants were free to spend as long as they wanted on the tasks; average completion time was around 25 minutes.

Procedure Participants began by reading a written design brief describing the client and their assigned task; see the

Appendix for the full text. Participants using d.tour also received a link to the d.tour Web interface.

Participants using d.tour watched a four-minute video tutorial which launched automatically when participants first viewed the tool and was available for repeat viewing at any time. For participants using search engines, we assumed familiarity with Google or a similar search engine.

Participants then searched for design examples. Upon finishing their search, they responded to a questionnaire. They first provided URLs and explanations for their chosen examples. They then answered survey questions about their search strategies and overall experience.

Results

Participants using search engines mainly reported querying with keywords such as 'science,' 'kids,' and 'education.' They also retrieved pages from their own mental search history, recalling well-designed, potentially relevant sites they had seen recently.

Several of these participants cited the lack of style-based search or the insufficiency of searching by page content as serious roadblocks. One participant wished for a search function that "rated sites by mood or feel." Another observed that "Google is only really useful for searching for actual page content, not page style. Google was less useful than I imagined it would be."

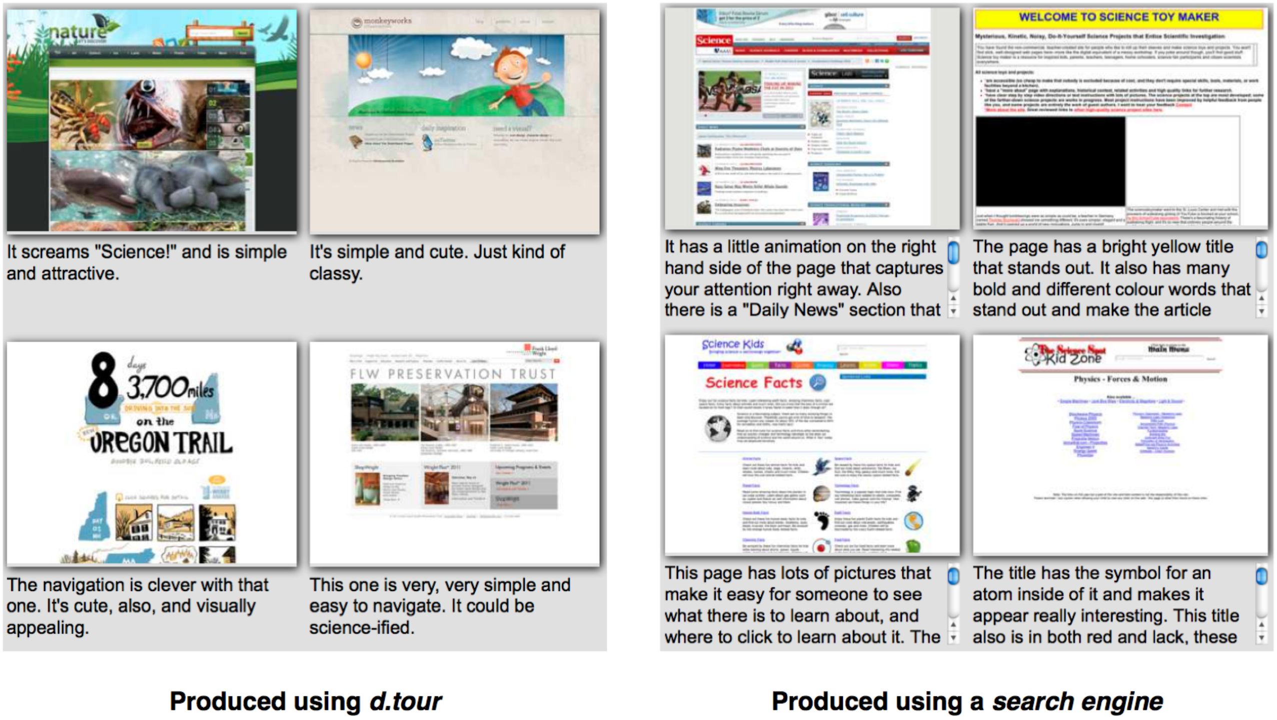
Feedback from participants using d.tour was generally positive. One participant described d.tour as a "really good tool to find quick examples." Every search technique was used by some participant, the most heavily used being 'show more like'. A participant mentioned that different query types proved useful at different stages of his search: "It was hard for me to describe what I liked . . . But once I had some examples . . . it made it easy to find common ground and then make the search more explicit." This account aligns well with the 'orienteering' theory of search described in [22].

Participants also singled out specific interface features as useful. One found value in having the 'Current Query' always present on the side of the screen as reminder of search history. Another reported using "a very zoomed-out view" to compare many designs simultaneously. The 'Ambient Style Display' received mixed feedback, as a few participants reported difficulty interpreting and operationalizing the information it provided.

Some participants also cited low corpus size as a limitation for d.tour; potential remedies for this issue were discussed earlier in the Implementation section.

Summary

Compared to those who used search engines, participants who used d.tour reported a positive example-finding experience. They identified key search features and interface components as specifically helpful in their exploration. These



It screams "Science!" and is simple and attractive.

It's simple and cute. Just kind of classy.

The navigation is clever with that one. It's cute, also, and visually appealing.

This one is very, very simple and easy to navigate. It could be science-ified.

It has a little animation on the right hand side of the page that captures your attention right away. Also there is a "Daily News" section that

The page has a bright yellow title that stands out. It also has many bold and different colour words that stand out and make the article

This page has lots of pictures that make it easy for someone to see what there is to learn about, and where to click to learn about it. The

The title has the symbol for an atom inside of it and makes it appear really interesting. This title also is in both red and lack, these

Produced using *d.tour*

Produced using a *search engine*

Figure 7: Inspiration Boards (examples and rationales) produced by participants. (Right) Many participants using search engines chose pages on a topic relevant to the design task. (Left) Participants using d.tour focused on the style and presentation of their chosen designs.

initial results suggest that style-based exploration of a design gallery can improve the example-finding process.

Further research is needed to isolate the factors that make d.tour effective. Specifically, it will be important to build a theory of how corpus size and interaction style affect user experience.

DISCUSSION

We showed the participants' Inspiration boards to two professional Web designers and two members of the Science Bus. Both groups strongly preferred pages that specifically addressed the Science Bus's content presentation needs—how to showcase recent work, for instance. This suggests that design search tools should support search by function in addition to style. Design pattern galleries, such as patterntap.com and designofsites.com/designpatterns, provide organized collections of functional Web page elements (navigation bars, slideshows, etc.) Unifying d.tour's high-level style search with pattern galleries' more bottom-up approach could markedly improve the search experience. To achieve this, 'show more like' could be refined to operate on individual page *components*; automatically classifying Web page components might provide a starting point [17].

One participant who used d.tour wished she could insert the Science Bus's content into candidate example pages to see how well it fit. While the examples found in our experiment were used for inspiration as-is, one could imagine integrating d.tour with the Bricolage Web page retargeting algorithm [18]. Seeing example pages adapted to the desired content could help non-experts look past surface dissimilarities and overcome conformity bias.

Participants who used d.tour produced more domain-diverse examples than those who used search engines. d.tour's currently small corpus makes focusing on a particular domain difficult, but even given a larger corpus, style-based search could still encourage this behavior more than keyword search. Search engines are optimized for topic-based search, so people often use them to find designs by domain. When searchers exhaust the good designs in one domain, however, finding a new one to explore can be challenging. Searchers often navigate between 'patches' of relevant content, continually trading off the cost of further examining one patch with the cost of locating a new one [23]. With keyword search, patches are topically organized, and textually-presented search results lack a scent/preview of the design. With d.tour, design patches are style-based, and 'show more like' makes jumping between domains easier. For example, when asked to show more like apple.com. d.tour retrieves onlynylives.com and clubjoomla.com amongst others—these design-similar pages come from three very disparate domains (Figure 1).

Style-based gallery exploration may have an advantage over search engines simply by presenting results in a design gallery. For design finding, visual display of search results provides stronger information scent [23] than the textual document surrogates [11] typically presented by Web search engines. This helps the searcher decide which results are worth clicking through. It also shrinks the gulf of evaluation [14], providing more immediate feedback on query usefulness. Though commercial search engines rely primarily on textual display, several research systems have explored visual result presentation [6, 31]. Some search engines such as Bing and Google provide optional visual previews, increasing their usefulness as design search tools. Gallery presentations are common for Web image search. For search tasks that are exploratory, low precision, or difficult to specify, galleries are highly effective.

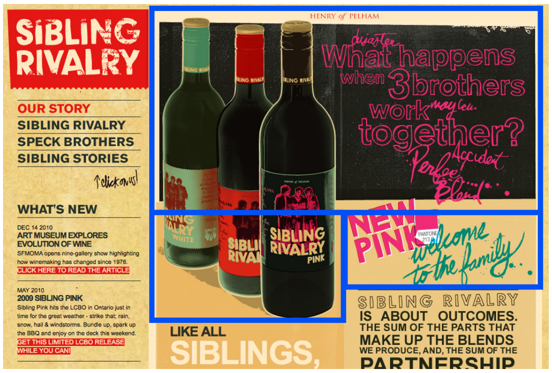
CONCLUSION AND FUTURE WORK

This paper presented a new approach for style-based design example-finding. It described d.tour, a prototype gallery exploration tool that supports several novel search interactions by extracting and analyzing design features from a curated corpus of Web pages. In an initial evaluation, d.tour received positive feedback from participants performing a design-example finding task.

There are several opportunities for future work. First, some stylistic properties likely influence human similarity judgments more than others. However, d.tour currently standardizes all page features, effectively giving them all the same weight. One could potentially learn the appropriate weights from human paired comparisons of Web pages; [29] proposes a related method for learning the perceptual distance between glossy materials.

Second, expanding the stylistic vocabulary is critical to supporting robust, language-based design search. As mentioned previously, one could apply machine learning techniques to a large corpus of stylistically-tagged Web pages to learn which design features predict frequently-occurring tags. Research is required to design a task environment in which crowds can produce high-quality style labels.

Third, an expanded style vocabulary will require more advanced design features. Many of d.tour's features rely on the HTML DOM, which simplifies a page to a hierarchy of rectangles. In the page shown in Figure 8, the wine bottles and some of the type overlap, breaking the strong grid lines specified by the DOM. d.tour's current features cannot detect 'organic' layouts where perceptually-coherent chunks span DOM elements. Combining DOM analysis with more



DEC 14 2010
ART MUSEUM EXPLORES EVOLUTION OF WINE
SFMOMA opens nine-gallery show highlighting how winemaking has changed since 1976.
CLICK HERE TO READ THE ARTICLE

MAY 2010
2009 SIBLING PINK
Sibling Pink hits the LCBO in Ontario just in time for the great weather - strike that; rain, snow, hail & windstorms. Bundle up, spark up the BBQ and enjoy on the deck this weekend.
GET THIS LIMITED LCBO RELEASE WHILE YOU CAN!

SIBLING RIVALRY IS ABOUT OUTCOMES. THE SUM OF THE PARTS THAT MAKE UP THE BLENDS WE PRODUCE, AND, THE SUM OF THE PARTNERSHIP

Figure 8: The wine bottles and the type on this page overlap and span multiple DOM nodes (outlined in blue). d.tour's current feature set cannot support queries about 'organic' page layouts such as this.

sophisticated computer vision algorithms could address this limitation.

Fourth, d.tour does not attempt to capture non-static aspects of design such as animation or interaction. Future work could address searching for designs by such elements.

Fifth, a full-fledged system should scale to much large page databases—perhaps even the entire Web. Such a system could use standard Web search indexing techniques to accelerate keyword queries. Locality Sensitive Hashing has been successfully applied to the nearest neighbor problem with large amounts of high-dimensional data [8]; it could help pick similar designs out of a large database.

Finally, and perhaps most exciting, is the possibility of embedding d.tour into real Web design software and observing how it affects the creative process.

ACKNOWLEDGMENTS

We would like to thank Martin Hunt for his help implementing the Web interface, Steven Dow for helping with study design, and the Stanford HCI group for valuable discussion. We would also like to thank Mira Dontcheva, Ranjitha Kumar, and the reviewers for their helpful feedback on earlier drafts of this paper. This work was supported by a Stanford Graduate Fellowship, the Stanford Computer Forum, and NSF Grant IIS-0745320.

REFERENCES

1. node.js. http://nodejs.org, 2011. Retrieved March 17, 2011.
2. The Science Bus. http://www.stanford.edu/group/sciencebus, 2011. Retrieved March 17, 2011.
3. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, June 2005.
4. J. Cui, F. Wen, and X. Tang. Intentsearch: interactive online image search re-ranking. In *Proceedings of MM: ACM Conference on Multimedia*. ACM, 2008.
5. M. Dixon and J. Fogarty. Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 2010.
6. S. Dziadosz and R. Chandrasekar. Do thumbnail previews help users make better relevance decisions about web search results? In *Proceedings of SIGIR: ACM Conference on Research and Development in Information Retrieval*. ACM, 2002.
7. G. Giacinto and F. Roli. Bayesian relevance feedback for content-based image retrieval. *Pattern Recognition*, 2004.
8. A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of VLDB: International Conference on Very Large Data Bases*, pages 518–529. Morgan Kaufmann Publishers Inc., 1999.
9. S. Gregor and I. Benbasat. Explanations from intelligent systems: theoretical foundations and implications for practice. *MIS Quarterly*, 23, December 1999.
10. Y. Hashimoto and T. Igarashi. Retrieving web page layouts using sketches to support example-based web design. *2nd Eurographics Workshop on Sketch- Based Interfaces and Modeling*, 2005.
11. M. A. Hearst. *Search User Interfaces*. Cambridge Univesity Press, 2009.
12. S. R. Herring, C. Chang, J. Krantzler, and B. P. Bailey. Getting inspired!: understanding how and why examples are used in creative design practice. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 2009.
13. V. Hollink, M. van Someren, and V. de Boer. Capturing the needs of amateur web designers by means of examples. In *Proceedings of the 16th Workshop on Adaptivity and User Modeling in Interactive Systems*, 2008.
14. E. L. Hutchins, J. D. Hollan, and D. A. Norman. Direct manipulation interfaces. *Human-Computer Interaction*, 1(4), Dec 1985.
15. M. Y. Ivory and M. A. Hearst. Statistical profiles of highly-rated web sites. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 2002.
16. D. G. Jansson and S. M. Smith. Design fixation. *Design Studies*, 12(1), 1991.
17. C. Kulkarni and S. R. Klemmer. Automatically adapting web pages to heterogeneous devices. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems Extended Abstracts*. ACM, 2011.
18. R. Kumar, J. O. Talton, S. Ahmad, and S. R. Klemmer. Bricolage: Example-based retargeting for web design. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 2011.
19. B. Lee, S. Srivastava, R. Kumar, R. Brafman, and S. R. Klemmer. Designing with interactive example galleries. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 2010.
20. J. Lou, K. Chen, and C. Lei. A collusion-resistant automation scheme for social moderation systems. In *Proceedings of CCNC: IEEE Conference on Consumer Communications and Networking*. IEEE Press, 2009.
21. J. Marks et al. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH: ACM Conference on Computer Graphics and Interactive Techniques*. ACM, 1997.
22. V. L. O'Day and R. Jeffries. Orienteering in an information landscape: how information seekers get from here to there. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 1993.
23. P. Pirolli. *Information Foraging Theory: Adaptive Interaction with information*. Oxford University Press, 2007.
24. K. Porkaew and K. Chakrabarti. Query refinement for multimedia similarity retrieval in mars. In *Proceedings of MM: ACM Conference on Multimedia*. ACM, 1999.
25. P. Pu and L. Chen. Trust building with explanation interfaces. In *Proceedings of IUI: ACM Conference on Intelligent User Interfaces*. ACM, 2006.
26. J. A. Ramos. Using TF-IDF to determine word relevance in document queries. In *Proceedings of ICML: International Conference on Machine Learning*, 2003.
27. T. Samara. *Design Elements: A Graphic Style Manual*. Rockport Publishers, 2007.
28. N. Tintarev. Explanations of recommendations. In *Proceedings of RecSys: ACM Conference on Recommender Systems*. ACM, 2007.

29. J. Wills, S. Agarwal, D. Kriegman, and S. Belongie. Toward a perceptual space for gloss. *ACM Transactions on Graphics*, 28, September 2009.

30. W. Wong, I. Oberst, S. Das, T. Moore, S. Stumpf, K. McIntosh, and M. Burnett. End-user feature labeling: a locally-weighted regression approach. In *Proceedings of IUI: ACM Conference on Intelligent User Interfaces*. ACM, 2011.

31. A. Woodruff, A. Faulring, R. Rosenholtz, J. Morrsion, and P. Pirolli. Using thumbnails to search the web. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. ACM, 2001.

32. T. Yeh, T. Chang, and R. C. Miller. Sikuli: using GUI screenshots for search and automation. In *Proceedings of UIST: ACM Conference on User Interfaces Software and Technologies*. ACM, 2009.

APPENDIX: Experimental Design Brief

Introduction

The Science Bus is an after-school science curriculum for 2nd through 5th grade children at the East Palo Alto Charter School (EPACS), a great school in a low-income part of the San Francisco Bay Area. The program is developed and taught by graduate and undergraduate students at Stanford University.

The leaders of The Science Bus program are not satisfied with their current Web site; they think it's kind of ugly and want to make it a lot nicer and more design-oriented. They want their new design to serve three purposes:

- Be a recruitment tool for Stanford students
- Present their awesome work to anyone who's interested
- Be a resource for similar science education programs across the country

Stylistically, they want the new page to scream "Science!" They're also fans of simplicity, so the new page shouldn't be too cluttered.

Your Task: Find inspiring design examples

Your task is to provide URLs for *four (4) Web pages* that are good design examples for The Science Bus. The Science Bus should be able to build a well-designed page that meets their needs by referring to your examples for inspiration.

For each example page that you provide, you'll also be asked to *explain what makes that page a good example*. In your explanations, you might want to refer to color scheme, layout, typography, or other elements of style. This is by no means an exhaustive list; feel free to discuss anything about the pages you select.

The pages you provide don't have to be perfect in all aspects. Each page being excellent in just one–color scheme, for instance–can make for a great set of design examples.