Pacific Graphics (2016)
E. Grinspun, B. Bickel, and Y. Dobashi (Editors)

Icon Set Selection via Human Computation

Lasse Farnung Laursen[1], Yuki Koyama[1], Hsiang-Ting Chen[2], Elena Garces[3], Diego Gutierrez[3], Richard Harper[4], and Takeo Igarashi[1]

[1]The University of Tokyo, Japan [2]University of Technology, Australia
[3]Universidad de Zaragoza, Spain [4]Social Shaping Research, England

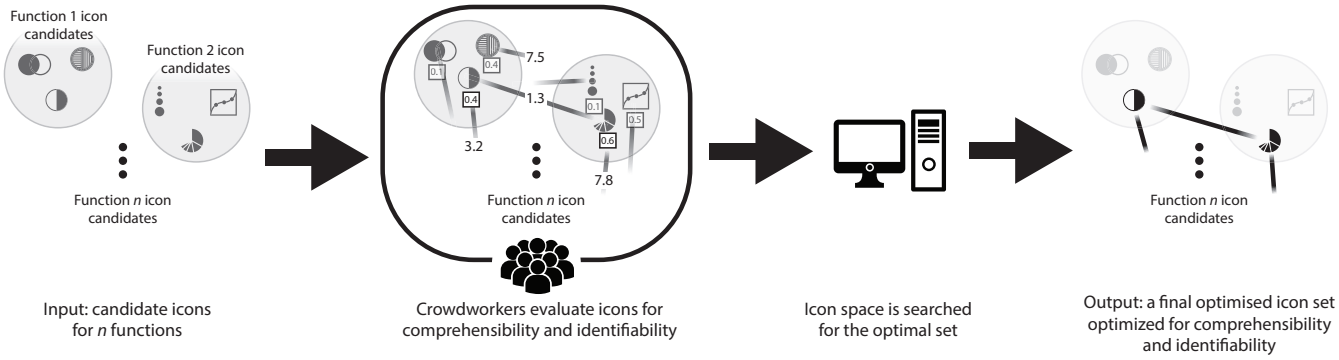


Figure 1: *An overview of our icon-set-selection method. For n functions requiring iconic representation, our method takes as input; n sets of icon candidates. All icon candidates, in each set, are assigned a comprehensibility and identifiability score using data gathered via human computation. Our method then automatically selects an icon set optimized for comprehensibility and identifiability.*

Abstract

Picking the best icons for a graphical user interface is difficult. We present a new method which, given several icon candidates representing functionality, selects a complete icon set optimized for comprehensibility and identifiability. These two properties are measured using human computation. We apply our method to a domain with a less established iconography and produce several icon sets. To evaluate our method, we conduct a user study comparing these icon sets and a designer-picked set. Our estimated comprehensibility score correlate with the percentage of correctly understood icons, and our method produces an icon set with a higher comprehensibility score than the set picked by an involved icon designer. The estimated identifiability score and related tests did not yield significant findings. Our method is easy to integrate in traditional icon design workflow and is intended for use by both icon designers, and clients of icon designers.

Categories and Subject Descriptors (according to ACM CCS): H.5.2. [Information Systems]: Information Interfaces and Presentation (e.g. HCI)—User Interfaces - Graphical user interfaces (GUI) I.3.8. [Computing Methodologies]: Computer Graphics—Applications

1. Introduction

Graphical user interface (GUI) design is a complex and challenging task. Many aspects of GUI design still rely on significant manual execution, such as icon selection, widget arrangement, color composition, etc. A significant body of research is dedicated to, or can be applied in service of, automating the GUI design process. For example, Fitts's law [Mac92] is often used to evaluate the performance of GUI elements, Xu et al. [XFIT14] and Gajos et al. [GW04] have proposed algorithms to automatically arrange UI elements. This paper attempts to tackle an important, yet underexplored, aspect of automating GUI design: icon selection.

Icons continue to be used in nearly all modern interfaces [Wie99, HBS15, CLKL16], since the emergence of WIMP ('Windows, icons, menus, pointers') systems by Xerox. Although previous research findings [Wie99, HBS15] highlight that stand-alone icons have no superiority over labels in terms of usability, they can provide information in a more compact space and have the potential to cut across language barriers [Böc96].

Automating the process of selecting useful functional icons is challenging because, despite the plethora of existing designs, the criteria for an icon's success is entirely dependent on its users. Selected icons will be used by a wide variety of people, each with different preferences, experiences, backgrounds, etc. This makes the right choices for selecting the ideal icon hard to distill for a single designer. The traditional approach to solving this problem is to employ user testing, in order to make a more informed decision. However, user testing is both time consuming and expensive.

The emergence of robust crowd sourcing platforms and human computation enable us to approach this problem in a different manner: measuring the end-user perspective via crowd-sourcing. Using crowd-sourcing to identify an optimal set of icons is not straightforward, as icons lack commonly measured performance properties of other interactive GUI design elements. For example, GUI widgets, such as sliders and menus, can be manipulated and evaluated via the resulting interaction. Icons do not inherently provide any kind of interactive functionality to measure. Therefore, we draw from existing icon design principles [HSC02] instead, and use comprehensibility and identifiability as our success criteria. Comprehensibility describes how well an icon communicates what it is intended to represent to the user. Identifiability refers to how visually distinguishable an icon is, i.e. how easily it is found among other icons. We measure both these properties, and use the data to produce a complete icon set optimized for the targeted end users.

2. Related Work

Human Computation is described by Von Ahn [VA05] as "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve". It is often applied in problems involving human perception (e.g [GSCO12, CKGF13, DBH14,GAGH14,HS12,KSI14,LHLF15,OLAH14] and our ours). In our paper we focus on these and similar works, but encourage readers to consult the broad overview of human computation based works presented by Quinn and Bederson [QB11].

Heer and Stone [HS12] use human computation to construct a probabilistic model of color names. Demiralp et al. [DBH14] present perceptual kernels: distance matrices for color, shape, and size derived from aggregate perceptual judgments. Donovan et al. [OLAH14] and Chaudhuri et al. [CKGF13] both use human computation and machine learning as a means to assign human-usable terms to fonts and visual content, respectively. Garces et al. [GAGH14] use human computation to cluster clip-art according to human visual perception; the authors later propose a method for efficient navigation and exploration of large clip-art data sets, taking into account both semantic information and style [GAHG16]. Liu et al. [LHLF15] apply machine learning to a database of categorized 3D furniture models (e.g. chair, table, etc.). Although our method generates data that could also help guide users, it is presented as a fully automated solution. The previous related works apply machine learning using a feature vector describing the input, allowing them to apply their methods without requiring further crowdsourced data. Due to the difficulty in extracting a useful feature vector from similarly styled icons, our machine learning relies solely on relative element comparisons. Another notable work using human computation is that of Koyama et al. [KSI14], who employ human computation to explore design parameter spaces that affect visual perception, such as photo filtering or 3d modeling. Both Koyama et al. and our work tightly integrate human computation into our methods and require new crowdsourced data for every use, contrary to the other works.

At the time of writing, we are unaware of other works utilizing crowd-sourcing to automate GUI design, nor are we aware of alternate methods of automating GUI icon selection. Thus we broaden our scope to include related works on **icon-design**, **-evaluation**, and **-generation**. An introduction to icon design and recommended practices is provided by Horton [Hor96, Hor94]. Our method, and evaluation thereof, is partially guided by the principles outlined in those works.

Nolan presents a precursor to our work, involving two studies comparing over 40 different icons [Nol89]. Performed in 1989, Nolan surveys over 350 participants, via letter correspondance, measuring 'appropriateness' and icon-meaning matching. Both 'appropriateness' and icon-meaning matching are ways of evaluating how comprehensible an icon is. His work shares similarities with ours, in how we evaluate our proposed method in our user study. Isherwood [IMC07] present more recent work focusing on how various icon characteristics (including comprehensibility) and prolonged exposure affects the speed and accuracy with which an icon is identified. Cherng et al. [CLKL16] demonstrate how electroencephalography (EEG) can serve as a tool to also evaluate icon properties.

A few notable works exist concerning icon generation for the purpose of improving the user interface. Lewis et al. [LRFN04] present a method which generates custom icon visuals derived from file-names. The visuals themselves are (by the authors own definition) generated arbitrarily and bear no resemblance to neither the file type, nor its contents. Kolhoff et al. [KPL08] expand upon this concept by generating (flower-like) icon visuals for music files that derives its shape from the audio content. Setlur et al. [SABAG*05] generate icons based on file-names using a stock photography database with tagged images. Contrary to our method these works concentrate on icons representing data (e.g. Text, Audio, Video, etc.), rather than functionality (e.g. Load, Save, Delete, etc.). Furthermore, these previous works either generate icons arbitrarily [LRFN04], or leverage meta-data [KPL08, SABAG*05] (e.g. file-name, data content, etc.) to generate icons, which is not naturally present for icons intended to represent functionality.

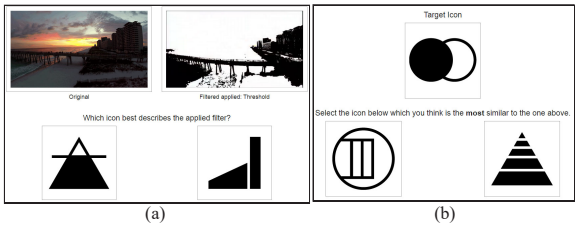
3. Icon Set Selection via Human Computation

Our method can be broken down into four parts, as depicted in Fig. 1. For the sake of clarity we use examples and icons from our test case (i.e. experimental setup) detailed in the following section. Our method first accepts n icon candidate groups as input. Each group represents a unique function or concept in an interface. Next, all icons within these groups are measured for comprehensibility and identifiability, using human computation. With the collected data, an exhaustive search is performed to finally produce an icon set optimized for comprehensibility and identifiability, consisting of n icons (one from each candidate group). In addition to comprehensibility and identifiability, another important factor in icon design is

style [HSC02]. Our method supports any visual style, as long as it is consistent for all $n \times m$ icons. A lack of consistent style will negatively influence the identifiability measurements, as crowd-workers will likely focus on stylistic differences, rather than differences between sign or symbol in the icon.

We denote an icon I_i^j with $i = 1...n$ functions (that the interface is to provide) and $j = 1...m$ candidates (for each function). For the sake of simplicity we let every icon candidate group contain m candidates, although our method easily supports a varying number of candidates per group. All of the $n \times m$ icons given as input are evaluated via crowd-sourcing. Workers are recruited and asked to perform relative comparisons between icons within a candidate group (comprehensibility) and in-between candidate groups (identifiability). Using the collected data, all icons are ranked for comprehensibility and embedded in euclidean space according to a dissimilarity measure. Finally, a complete icon set, optimized for both identifiability and comprehensibility, is found via a simple exhaustive search. Further details are provided in the following sections.

3.1. Human computation



(a) (b)

Figure 2: *Cropped screenshots from our human computation crowd-sourcing tasks. (a) A sample comprehensibility related task presented to crowd-workers. On top the original and filtered sample image is provded along with the filter name (Threshold). The text presented to the user reads: "Which icon best describes the applied filter?" (b) A sample identifiability related task presented to crowd-workers. The text presented to the user reads: "Select the icon below which you think is the most similar to the one above.".*

We use crowd-workers to approximate how comprehensible and identifiable each icon candidate is. An example task to measure comprehensibility and identifiability is shown in Fig. 2.a and Fig. 2.b, respectively.

Measuring **comprehensibility** relies solely on direct within candidate group comparisons (I_i^j, I_i^l where $j \neq l$). As Fig 2.a shows, each comprehensibility task consists of a visualization of the function or concept we seek to represent, and two icon candidates. Previous works involving icon evaluation rely primarily on stand-alone labels [Nol89, IMC07, CLKL16]. However, given the remote nature of crowd-sourcing, we opted to also include a before/after image describing the intended icon functionality (as shown in Fig 2.a). Alternatively, a short video is also a feasible option given the capabilities of contemporary web-browsers. In our case, the functionality we communicate is a visual filter, along with two icon candidates, each representing the same visual filter. The worker is asked to pick the icon they feel is the better representation. This pair-wise comparison enables us to establish rank in-between icons, in order of perceived comprehensibility.

How **identifiable** an icon is, is dependant on the other icons displayed alongside it, and therefore relies on inter candidate group comparisons (I_i^j, I_k^l where $i \neq k$). In each identifiability task we ask the crowd-worker to pick which icon of two candidates from one group, is most similar to an icon candidate from a different group, as shown in Fig 2.b. In accordance with Maaten and Weinberg [VDMW12] we assume the existence of a dissimilarity function $dist(I_i^j, I_k^l)$, which measures visual dissimilarity. We approximate this unknown function $dist()$ by collecting triplets, via the aforementioned relative comparisons [SJ04]:

$$\mathcal{T} = \{(I_i^j, I_k^a, I_k^b) \,|\, I_i^j \text{ is more similar to } I_k^a \text{ than } I_k^b\}. \quad (1)$$

A given triplet $(I_i^j, I_k^a, I_k^b) \in \mathcal{T}$ implies $dist(I_i^j, I_k^a) < dist(I_i^j, I_k^b)$, and with sufficient worker judgments, an acceptable approximation of $dist()$ is expected.

3.1.1. Quality control methods to detect invalid workers

Poor human computation results can often be caused by improperly designed tasks [KNB*13], as opposed to lazy or deceitful workers. However, some workers may be motivated primarily by financial benefits [EdV11, Ipe10, DHSC10] and attempt to finish tasks quickly, rather than accurately [CB09, KCS08]. We inject superfluous creative tasks requiring workers to describe an icon with a few simple sentences. This ensures a grasp of the English language and minimizes malicious behavior [EdV11]. The creative tasks also increase the amount of context switching which is actually preferable in crowd-sourcing [DRPC15] and prevents workers for habitually performing the same tasks as fast as possible. Finally, we duplicate tasks to ensure consistency and discard measurements when worker results fall below 70% consistency.

3.2. Icon set optimization

Our goal is to select multiple icons, forming a complete icon set optimized for comprehensibility and identifiability. A complete icon set $\mathcal{S}$ consists of n members ($\mathcal{S} = \{I_1, I_2, I_3, ..., I_n\}$), one for each function (, in our case filter), as depicted in Fig. 3.

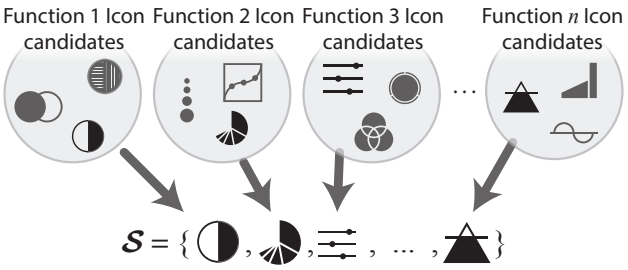


Figure 3: *n icon candidate groups, each contributing a single candidate icon to a complete icon set $\mathcal{S}$.*

We assign each complete set a score consisting of both comprehensibility $f_{\text{comp}}(\mathcal{S})$ and identifiability $f_{\text{ident}}(\mathcal{S})$ (adjusted by a

weight parameter ϕ) which we seek to maximize:

$$\begin{aligned} \underset{\mathcal{S}}{\text{maximize}} \quad & f_{\text{score}}(\mathcal{S}) = f_{\text{comp}}(\mathcal{S}) + \phi \times f_{\text{ident}}(\mathcal{S}), \\ \text{where} \quad & f_{\text{comp}}(\mathcal{S}) = \frac{1}{n} \sum_{I \in \mathcal{S}} \text{comp}(\, I \,), \\ & f_{\text{ident}}(\mathcal{S}) = \min(\, D(\mathcal{S}) \,), \\ & D(\mathcal{S}) = \{\, dist(I_a, I_b) \mid (I_a, I_b) \in \mathcal{S}, a \neq b \,\}. \end{aligned} \tag{2}$$

The parameter ϕ allows the user to specify whether to prioritize comprehensibility or identifiablity throughout the set selection process. The function $comp()$ returns a ranking score determined by the Bradley-Terry model, as described by Caron and Doucet [CD12]. A recent work expanded upon the Bradley-Terry model, to create Crowd-BT [CBCTH13], a more complex model designed specifically for crowdsourced data. However, for our approach the original version will suffice. The previously described function $dist()$ returns the euclidean distance between two icons calculated using stochastic triplet embedding as described by Maaten and Weinberger [VDMW12]. Since we are interested in maximizing the perceived visual difference between all icons in our complete set, we use the smallest distance (within a set of distances) as an indicator for how visually distinct the complete icon set is.

We maximize $f_{\text{score}}(\mathcal{S})$ by performing an exhaustive search of the complete icon set space, which has a run-time of $\mathcal{O}(m^n)$. For our application this took approximately 3 minutes. For sufficiently large values of m or n, this run-time becomes computationally intractable. A lower run-time could be achieved by applying an approximated combinatorial optimization algorithm, or probabilistic meta-heuristic (such as simulated annealing).

4. Experimental Setup

We apply our method to select icons for a new touch-based GUI used to create live visual performances. This scenario is appealing, as a standardized iconography has not been established for visual filters. Contemporary visual performance software (ArKaos GrandVJ, Resolume 4, Modul8, VDMX5) rely primarily on labels and visual previews. Our new interface is intended for novices, but should also support more experienced users. A key feature of the interface is to apply visual filters to real-time video. In order to determine the most popular visual filters used in live performances, we conducted interviews with visual performance artists and gathered data via surveys. The nine most popular visual filters (out of ∼150) were determined to be: brightness, contrast, gaussian blur, grayscale/saturation, hue, solarize, threshold, trails, and zoom. To mimic a typical design scenario, and ensure a consistent icon style, we hired a single designer to create five icon candidates (as suggested by Horton [Hor94]) for each of the nine visual filters. All 45 icons (depicted in the supplemental material) were designed with a simple silhouetted style as recommended by Horton [Hor94], given the early stages of our overall GUI design.

4.1. Implementation

We use Crowdflower (`crowdflower.com`) to gather human computation data. Crowdflower provides limited options for individually tailoring tasks for workers. Therefore, we merely use it as a means to recruit workers and redirect them to an external survey, hosted on our own server. Our goal is to collect comprehensibility and identifiability data for all 45 icons (five icon candidates for each of our nine visual filters).

Each icon candidate group contains $\binom{m}{2}$ unique icon pair combinations (for our application where $m = 5$, we have 10). Therefore, to exhaustively measure comprehensibility we must sample $n \times \binom{m}{2}$ relative comparisons (90 in our case). To exhaustively measure identifiability we must sample $(n \times m) \times (n-1) \times \binom{m}{2}$ relative comparisons, as one icon I_i^j must be compared with combinations from all other icon candidate groups. In our case, where $n = 9$ and $m = 5$, we require $45 \times (9-1) \times 10 = 3600$ comparisons. We assign each worker 10 unique identifiability tasks and 5 unique comprehensibility tasks, all in randomized order. We intentionally over-sample comprehensibility in order to maintain a high task diversity. After duplicating comprehensibility and identifiability tasks, and inserting context-switching creative tasks, the total number of tasks per worker is 39. Prior to commencing the tasks, each worker is given a brief tutorial (detailed in the supplemental material), including examples to minimize misunderstandings and clarify ambiguous terms. A minimum of 7 judgments (requiring ∼2500 valid workers) were collected for all comparisons. The entire process took approximately three days, and cost about 500 USD in total.

We generated several different icon sets and experimented with the adjustable parameter ϕ in order to maximize diversity in terms of both comprehensibility and identifiability. All of the generated sets are shown in Fig. 4, along with two icon sets optimized exclusively for either comprehensibility or identifiability. Through empirical testing we determined a good dimensionality setting for the stochastic triplet embedding to be $dims = 3$. More advanced methods to determine the optimal dimensionality exist, such as cross-validation.

5. Evaluation

We conducted a user study that compared the comprehensibility and identifiability between four complete icon sets produced by our method and one complete icon set chosen by the original icon designer. These five icon sets are denoted in Fig. 4 by having a shaded background. Among the four generated sets, two sets were optimized for only identifiability/comprehensibility and two sets covers the ϕ spread of 0.2-0.5 and 0.7-1.0 to maximize variety.

The study tested the comprehensibility and identifiability of these sets via a between-subjects crowdsourced setup. Two distinct series of tasks were used:

- To test **comprehensibility**, we present each participant with a representation of one of our nine visual filters and a complete icon set below, as shown in Fig. 5.a. The participants are then asked to pick the icon they feel best represents the filter. They are explicitly informed that they can pick the same icon multiple times, for different filters. This task is repeated once for every filter. The icon set order is randomized for every user, but kept in that same order for the duration of the nine tasks, so the user can rely on spatial awareness to avoid unintentionally selecting the same icon multiple times.

Parameter Settings	Icon Set	$f_{\text{comp}}(\mathcal{S})$	$f_{\text{ident}}(\mathcal{S})$	% Correct picks Comprehensibility	% Correct finds Identifiability	Time to find in secs (+ SD) Identifiability
Designer's Set		0.259	0.010	36.90%	99.34%	1.69 (0.96)
Only Comprehensibility		0.350	0.031	38.99%	99.29%	1.67 (0.92)
$\phi = 0.1$		0.345	0.087	-	-	-
$0.2 \leq \phi \leq 0.5$		0.296	0.436	34.75%	99.87%	1.58 (0.93)
$\phi = 0.6$		0.239	0.541	-	-	-
$0.7 \leq \phi \leq 1.0$		0.228	0.557	31.57%	99.37%	1.58 (0.90)
$10 \leq \phi \leq 1000$		0.168	0.611	-	-	-
Only Identifiability		0.124	0.611	28.30%	99.11%	1.58 (0.94)

Brightness Contrast Gaussian Blur Grayscale/ Saturation Hue Solarize Threshold Trails Zoom

Figure 4: *A hand picked icon set, chosen by the original icon designer, and icon sets produced via our method with varying values for ϕ. Each row shows the $f_{\text{comp}}(\mathcal{S})$ and $f_{\text{ident}}(\mathcal{S})$ score as calculated by our method. Percentage of correct picks/finds from the comprehensibility and identifiability user study tasks are shown on the right, along with the measured average time and standard deviation to find an icon in a given set. The rows containing the five icon sets tested as part of our user study have a shaded background.*

- To test **identifiability**, we show each worker one of the nine icons on screen and ask them to memorize it, as depicted in Fig. 5.b. On the following screen the users are asked to pick the icon they just memorized as quickly as possible, among the complete icon set as shown in Fig. 5.c. This task is repeated once for each of the nine icons, and the complete set is always shown in a randomized order.

In both tasks, we present icons on a line to minimize inter-icon effcts [CLKL16]. We measure time-to-completion and success rate in both tasks.

We implemented control methods to discard invalid users in the user study, similar to those used during the human computation tasks. We duplicate both series of tasks, but do not intermingle them. A user will always be presented with nine identifiability/comprehensibility tasks in a row. The previously described creativity tasks are only placed before or after a series of nine identifiability/comprehensibility tasks. Additionally, we relax the comprehensibility consistency constraint to 30%, as is it unreasonable to expect the user to remember every previous choice they've made if the icons make no sense to them. Conversely we increase the identifiability consistency constraint to 80% as finding one icon among nine others, designed to be visually distinct, is expected to be a straight-forward task. We discard data if the user makes more than 30% incorrect choices during the identifiability task, or picks fewer than four distinct icons during the comprehensibility task. We also disregard results where the reported times were either negative, or more than ten seconds during either of the identifiability tasks. We solicited crowd-workers from both Crowdflower and Microworkers (`microworkers.com`) in order to quickly collect a minimum of 50 valid studies per icon set.

5.1. Results and discussion

The result of the user study, and the estimated comprehensibility and identifiability scores, are shown in Fig. 4. Each row in the figure represents an icon set selected using our method, apart from the icon set in the top row, which was handpicked by the original icon designer. The five icon sets tested in our user study are indicated with a shaded background row. These rows containing tested

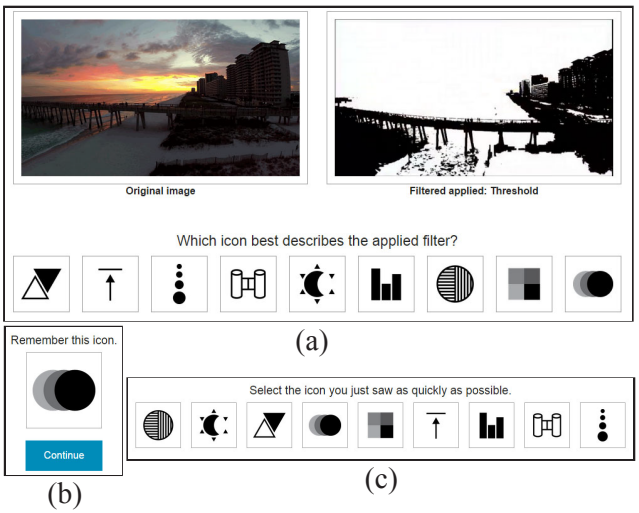


Figure 5: *Cropped screenshots from our user study tasks. (a) A sample comprehensibility evaluation task presented to crowd-workers. (b,c) Part one and two of a sample identifiability evaluation task presented to crowd-workers.*

icon sets also show the percentage of correct picks made during the comprehensibility and identifiability tasks. Time-to-completion is also shown for the identifiability tasks.

We compare our estimated comprehensibility and identifiability scores with the results from our user study. There is a correlation between the comprehensibility score as determined by our method and the percentage of correctly picked icons (during comprehensibility tasks) from a given set ($R^2 \approx 0.923$). The results pertaining to identifiability did not yield significant results. The near perfect percentage of correctly found icons for each set during identifiability tasks, points to success rate not being a useful indicator of identifiability. Although the time to find an icon also correlates with our identifiability score, the high standard deviation (of approximately one second) renders this result insignificant.

Comparing the estimated comprehensibility score and user study comprehensibility test results, indicates that icon sets picked using our method have a similar or higher comprehensibility score as one picked by a designer. For functionality without a well established iconography, or GUI's with a high amount of functionality to represent, our method becomes particularly helpful. Without past signs or symbols to rely on, or with an increasing number of hard-to-quantify user preferences, manually picking the best icons becomes very difficult. Using our method, the icon designer can be left to focus on simply coming up with as many representative icons as possible.

6. Acknowledgments

We thank the reviewers, participating VJs (Adam Kendall, Ana Carvalho, Benton C Bainbridge, and Oli Sorenson), crowdworkers, and Iliyan Nachev who designed all the icon candidates. This work was supported by JSPS KAKENHI Grant Number 24-02734, and a generous donation from MSRA. Photo 'Sunset at the Pier' by Mike Haytack, CC BY 3.0.

References

[Böc96] BÖCKER M.: A multiple index approach for the evaluation of pictograms and icons. *Computer Standards & Interfaces 18*, 2 (1996). 1

[CB09] CALLISON-BURCH C.: Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proc. EMNLP '09* (2009), ACL, pp. 286–295. 3

[CBCTH13] CHEN X., BENNETT P. N., COLLINS-THOMPSON K., HORVITZ E.: Pairwise ranking aggregation in a crowdsourced setting. In *Proc. WSDM '13* (2013), ACM, pp. 193–202. 4

[CD12] CARON F., DOUCET A.: Efficient bayesian inference for generalized bradley–terry models. *J. Comput. Graphical Statistics 21*, 1 (2012), 174–196. 4

[CKGF13] CHAUDHURI S., KALOGERAKIS E., GIGUERE S., FUNKHOUSER T.: Attribit: content creation with semantic attributes. In *Proc. UIST '13* (2013), ACM, pp. 193–202. 2

[CLKL16] CHERNG F.-Y., LIN W.-C., KING J.-T., LEE Y.-C.: An eeg-based approach for evaluating graphic icons from the perspective of semantic distance. In *Proc. CHI '16* (2016), ACM. 1, 2, 3, 5

[DBH14] DEMIRALP C., BERNSTEIN M. S., HEER J.: Learning perceptual kernels for visualization design. *IEEE Trans. InfoVis. 20*, 12 (2014), 1933–1942. 2

[DHSC10] DOWNS J. S., HOLBROOK M. B., SHENG S., CRANOR L. F.: Are your participants gaming the system?: screening mechanical turk workers. In *Proc. CHI '10* (2010), ACM, pp. 2399–2402. 3

[DRPC15] DAI P., RZESZOTARSKI J. M., PARITOSH P., CHI E. H.: And now for something completely different: Improving crowdsourcing workflows with micro-diversions. In *Proc. CSCW '15* (2015), ACM, pp. 628–638. 3

[EdV11] EICKHOFF C., DE VRIES A.: How crowdsourcable is your task. In *Proc. CSDM '11 at WSDM '11* (2011), pp. 11–14. 3

[GAGH14] GARCES E., AGARWALA A., GUTIERREZ D., HERTZMANN A.: A similarity measure for illustration style. *ACM Trans. Graph. 33* (2014), 93:1–93:9. 2

[GAHG16] GARCES E., AGARWALA A., HERTZMANN A., GUTIERREZ D.: Style-based exploration of illustration datasets. *Multimedia Tools and Applications* (2016), 1–20. 2

[GSCO12] GINGOLD Y., SHAMIR A., COHEN-OR D.: Micro perceptual human computation for visual tasks. *ACM Trans. Graph. 31*, 5 (2012). 2

[GW04] GAJOS K., WELD D. S.: Supple: automatically generating user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces* (2004), ACM, pp. 93–100. 1

[HBS15] HUANG S.-C., BIAS R. G., SCHNYER D.: How are icons processed by the brain? neuroimaging measures of four types of visual stimuli used in information systems. *Journal of the Association for Information Science and Technology 66*, 4 (2015), 702–720. 1

[Hor94] HORTON W. K.: *The Icon Book: Visual Symbols for Computer Systems and Documentation*. John Wiley & Sons, Inc., New York, NY, USA, 1994. 2, 4

[Hor96] HORTON W.: Designing icons and visual symbols. In *CHI '96 Course Notes* (1996), ACM, pp. 371–372. 2

[HS12] HEER J., STONE M.: Color naming models for color selection, image editing and palette design. In *Proc. CHI '12* (2012), ACM, pp. 1007–1016. 2

[HSC02] HUANG S.-M., SHIEH K.-K., CHI C.-F.: Factors affecting the design of computer icons. *Int. J. Industrial Ergonomics 29*, 4 (2002), 211–218. 2, 3

[IMC07] ISHERWOOD S. J., MCDOUGALL S. J., CURRY M. B.: Icon identification in context: The changing role of icon characteristics with user experience. *Human Factors: The Journal of the Human Factors and Ergonomics Society 49*, 3 (2007), 465–476. 2, 3

[Ipe10] IPEIROTIS P. G.: Demographics of mechanical turk. 3

[KCS08] KITTUR A., CHI E. H., SUH B.: Crowdsourcing user studies with mechanical turk. In *Proc. CHI '08* (2008), ACM, pp. 453–456. 3

[KNB*13] KITTUR A., NICKERSON J. V., BERNSTEIN M., GERBER E., SHAW A., ZIMMERMAN J., LEASE M., HORTON J.: The future of crowd work. In *Proc. CSCW '13* (2013), ACM, pp. 1301–1318. 3

[KPL08] KOLHOFF P., PREUSS J., LOVISCACH J.: Content-based icons for music files. *Computers & Graphics 32*, 5 (2008), 550–560. 2

[KSI14] KOYAMA Y., SAKAMOTO D., IGARASHI T.: Crowd-powered parameter analysis for visual design exploration. In *Proc. UIST '14* (2014), ACM, pp. 65–74. 2

[LHLF15] LIU T., HERTZMANN A., LI W., FUNKHOUSER T.: Style compatibility for 3d furniture models. *ACM Trans. Graph. 34*, 4 (2015), 85:1–85:9. 2

[LRFN04] LEWIS J. P., ROSENHOLTZ R., FONG N., NEUMANN U.: VisualIDs: Automatic Distinctive Icons for Desktop Interfaces. *ACM Trans. Graph. 23*, 3 (2004), 416–423. 2

[Mac92] MACKENZIE I. S.: Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction 7* (1992). 1

[Nol89] NOLAN P. R.: Designing screen icons: Ranking and matching studies. *Proc. HFES '89 33*, 5 (1989), 380–384. 2, 3

[OLAH14] O'DONOVAN P., LĪBEKS J., AGARWALA A., HERTZMANN A.: Exploratory font selection using crowdsourced attributes. *ACM Trans. Graph. 33*, 4 (2014), 92. 2

[QB11] QUINN A. J., BEDERSON B. B.: Human computation: a survey and taxonomy of a growing field. In *Proc. CHI '11* (2011), ACM, pp. 1403–1412. 2

[SABAG*05] SETLUR V., ALBRECHT-BUEHLER C., A GOOCH A., ROSSOFF S., GOOCH B.: Semanticons: Visual metaphors as file icons. In *Computer Graphics Forum* (2005), vol. 24, Wiley Online Library, pp. 647–656. 2

[SJ04] SCHULTZ M., JOACHIMS T.: Learning a distance metric from relative comparisons. *Proc. NIPS '03* (2004), 41–48. 3

[VA05] VON AHN L.: *Human Computation*. PhD thesis, 2005. 2

[VDMW12] VAN DER MAATEN L., WEINBERGER K.: Stochastic triplet embedding. In *Proc. MLSP '12* (2012), IEEE, pp. 1–6. 3, 4

[Wie99] WIEDENBECK S.: The use of icons and labels in an end user application program: An empirical study of learning and retention. *Behaviour & Information Technology 18*, 2 (1999), 68–82. 1

[XFIT14] XU P., FU H., IGARASHI T., TAI C.-L.: Global beautification of layouts with interactive ambiguity resolution. In *Proc. UIST '14* (2014), ACM, pp. 243–252. 1