EUROGRAPHICS 2005 / M. Alexa and J. Marks
(Guest Editors)

Volume 24 (2005), Number 3

Semanticons: Visual Metaphors as File Icons

Vidya Setlur Conrad Albrecht-Buehler Amy A. Gooch Sam Rossoff Bruce Gooch

Northwestern University

Abstract

Semanticons can enhance the representation of files by offering symbols that are both meaningful and easily distinguishable. The semantics of a file is estimated by parsing its name, location, and content to generate a 'context', which is used to query an image database. The resulting images are simplified by segmenting them, computing an importance value for each segmented region, and removing unimportant regions. The abstract look-and-feel of icons is achieved using non-photorealistic techniques for image stylization. We increase the effectiveness of the semanticons by compositing them with traditional and familiar interface icons. Two psychophysical studies using semanticons as stimuli demonstrate that semanticons decrease the time necessary to locate a file in a visual search task and enhance performance in a memory task.

Categories and Subject Descriptors (according to ACM CCS): H.1.2 [Information Interfaces and Presentation]: Human Factors I.3.8 [Computer Graphics]: Applications I.3.3 [Computer Graphics]: Display Algorithms

1. Introduction

Graphical user interface icons serve as pictographic representations of files' content or purpose. In current desktop file system interfaces, some files have a natural visual representation, such as thumbnails used for image files, while others are assigned icons by applications that generate the files. Assigning icons solely by application may lead to rows of identical icons that are neither physically nor perceptually distinctive. Physical distinctiveness means that the icons must be visually distinguishable. Perceptual distinctiveness refers to the viewer's understanding of what the icon represents, or what we refer to as the *semantics* of the icon. The goal of our work is to automatically generate icons that better reveal the semantics of desktop file contents. The resulting *semanticons* can enhance the representation of files in a Graphical User Interface (GUI) by providing semantically and graphically distinguishable symbols.

2. Background

Previous work has investigated features that contribute to recognizable icons. Moyes [Moy94] provides evidence that subjects associate an icon with a command by either its position in the interface or its shape, but not both. Whether the user relies on position or shape seems to depend on whether the shapes of icons are easy or difficult to learn. Byrne [Byr93] suggests that simple icons can be distinguished by a few features, while complex icons are no better than simple rectangles. Woodruff *et al.* created thumbnail images of web pages enhanced with the pages' contents to improve visual search tasks [WFR*01]. Suh et al. [SLBJ03] proposed automatic thumbnail cropping based on a visual attention model to detect interesting areas in an image.

Russell and Dieberger [RD03] demonstrated a design pattern based method that creates visual compositions to summarize large collections of media. The work aims to make the task of finding a target image easier in the collection of images. Multidimensional icons [HH90] are an attempt to convey file content by projecting a different representation of the file on to each side of a cube. While multidimensional icons provide additional meaning for a file icon, the rotating cube depiction allows only three views to be immediately visible at a time. Multidimensional icons tend to provide more insight into the file's type and usage, and less into the file's content.

Recently, Lewis et al. [LRFN04] created automatic distinctive icons, called VisualIDs, to augment the graphical display of files with "visual scenery." VisualIDs aid recognition by employing both an orderly graphical layout and distinctive appearance. We build on this idea by providing the user with a more recognizable image as its unique iden-

Figure 1: Semanticons *generated by our system for various filenames.*

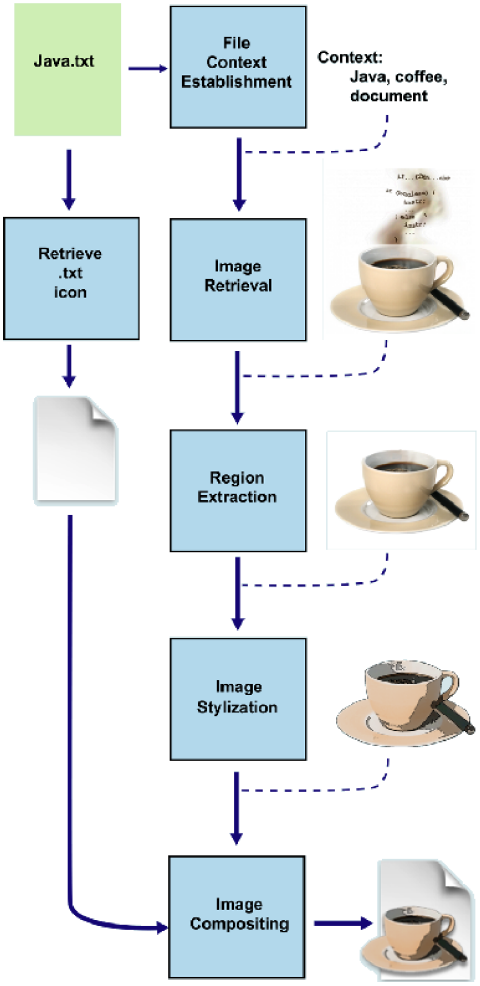

Figure 2: *Outline of Semanticon Generation Process.*

tifier, and then expand on this idea further by using the image to reveal the actual content of the file.

3. Semanticon Creation Process

To describe the semanticon creation process, we define *term* to be any word or phrase, and *context* to be a set of terms obtained by parsing the name, path, and textual content of a file. Our method automatically generates semantically enhanced icons in five steps. We first establish the context of file. Next, we use the context to retrieve images from a stock photography database, then extract the important regions of the image, stylize the image, and finally composite the results to generate a *semanticon*. Figure 2 illustrates this process.

3.1. File Context Establishment

A meaningful icon uses imagery that is either literally or figuratively connected to the file's content or purpose [RK88]. To find imagery that makes this connection, we need to establish a context for the file. Our system uses the context as a query to a database of images tied to keywords. We generate a collection of terms to use as the context of the file.

Generating context using filenames

Extracting a set of terms from a filename is challenging because files are usually named in a form of shorthand. Users often employ abbreviations and word separators, such as capital letters, hyphens, spaces, underscores, or periods to shorten filenames. Examples of such abbreviations are "sys" for system, "chp" for chapter, and examples of filenames using word separators are "java-tomcat.txt", "softwareProcesses.xls". In order to create context we need to translate this shorthand back into its unabridged form. We begin this translation process by splitting the filename at each word separator into segments, which we call *tokens*. These tokens may represent abbreviations, full English words, or a combination of English words. We define the result of dividing a token into one or more substrings as a *split*. Our parsing algorithm generates all possible splits for each token to determine the set of English words that the token represents. For example, some possible splits for the token "accessfwd" might be:

- "access", "fwd"
- "acc", "ess", "fwd"
- "acce", "ss", "fwd"

laptop/article.rtf java/article.rtf mailing-list-faq/article.rtf firewall/article.rtf committers/article.rtf

Figure 3: Semanticons for article.rtf stored in various file locations. In cases where files names are similar, the parent directory influences the semanticon creation.

We then rank the splits according to the ratio of recognized substrings to the total number of substrings in the split, and choose the highest ranked split for each token. If more than one split has the same ranking, we choose the split that contains the longest substring. A substring in each split is considered recognizable if any of the following rules are applicable:

1. The substring itself is a valid English word.
2. The substring is three characters long, and is a commonly used abbreviation [AL98]. (e.g. "dev" for "development", "lib" for "library", "fwd" for "forward")
3. The substring consists of consecutive consonants, and is a valid English word for a combination of vowels interspersed between the consonants. (e.g. "dbg" for "debug")

A parsing grammar verifies the validity of an English word by checking whether it is an entry in an online dictionary corpus. However, we ignore the 500 most common terms in the English language (e.g. "I", "the", "and", "there") because they are non-content words.

By applying the parsing rules to the above "accessfwd" example, the two substrings "access" and "fwd" in the first split represent a valid English word, "access" and a known abbreviation for "forward". So the rank of the split is $2/2 = 1$. In the second example, "acc" resolves to valid words of which we choose the first word (e.g. "accelerate", "accent", "access", etc.), "ess" resolves to a valid word (e.g. "essence", "essential", etc.), and "fwd" resolves to a valid English word "forward." The rank of this split is $3/3 = 1$. The third split has two invalid words, "acce", "ss", and one valid word "forward" for "fwd." The rank of the third split is $1/3 = 0.33$. We select the first split because it has the longest substring ("access"), even though the first and second splits are ranked equally.

Our system can infer additional context by considering the parent directories as part of the filename. If the complete file path is provided with the file, our system considers the immediate directory which contains the file and ignores the rest of the file path. For example, consider a user who has several files named *article.rtf* stored in different directories. Semanticons generated for each file have the common visual representation of a text file, but each is distinctive due to variations in directory structure as shown in Figure 3.

Generating context using file content

The file's content is useful for generating semanticons when the user determines that filename does not reflect the file's actual content. Our system determines the context from the file's textual content using established information retrieval methods [SWY75]. We begin by removing most common English words as described in the previous method. Next we extract the most frequently occurring terms. In practice, we have found that the first 20% of the most frequent terms sufficiently represents the contents of the document for the purpose of generating 4 semanticons per file. Our system then identifies frequent noun phrases in the text (e.g. "White House") to further refine the context. The frequent terms and noun phrases together form the context and are used to generate a query to find corresponding images.

When using the filename to determine the context, a student essay about "myth" and the "police" named "Frosh7.txt", produces a semanticon of a freshman. On the other hand, a semanticon generated from the file's content, includes images of law enforcement and mythical objects. (Figure 4).

Figure 4: Semanticons for 'Frosh7.txt'. Left: Semanticon generated from filename. Right: Semanticon generated from file content. The file is an essay about myths surrounding police officers.

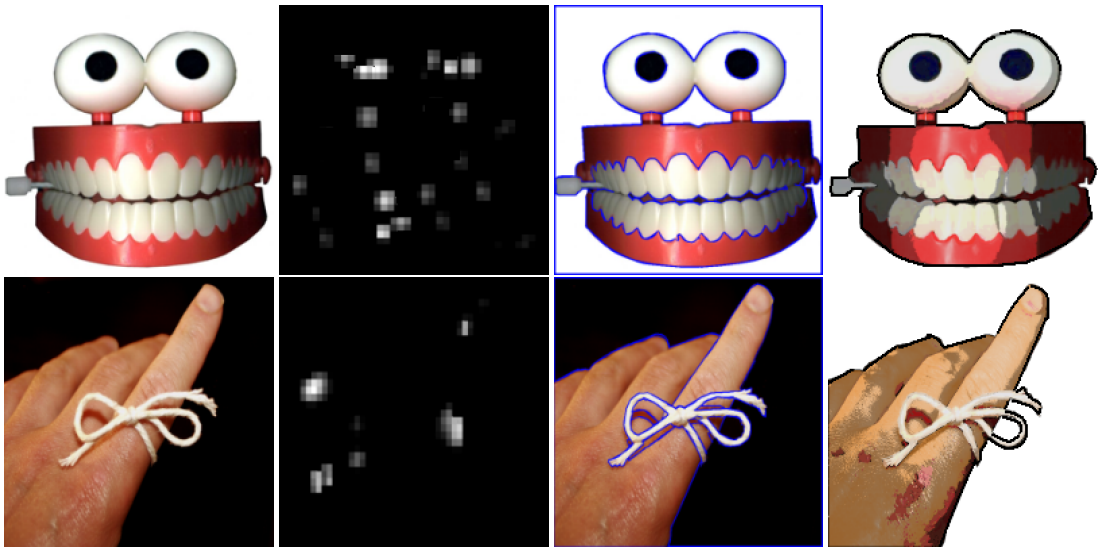

Figure 5: Region extraction and image stylization. From left to right: Original image, importance map, segmented image, cartooning.

3.2. Image Retrieval

Our semantically-guided information retrieval process automatically generates queries for retrieving images from a stock photography database. We choose a stock photography warehouse as our image database [Ind]. This service provides a list of tightly coupled keywords associated with each image. The images typically are high-quality photographs and illustrations, have very few subjects, and often have uniform, neutral backgrounds, which simplifies our image extraction process. We also provide users with the option of including their own images in place of, or in conjunction with, the images returned by the service. Fogarty et al. [FFH01] query a stock photography database to create a collage representing contents of e-mails. However, our work focuses on using stock photography images to represent file icons.

We initially considered using only clipart for generating semanticons because clipart is often semantically meaningful and iconic in nature. However, publicly accessible databases of clipart are typically smaller than stock photography databases. The larger stock photography database provides us with a greater variety of images available to our system and thereby decreasing the likelihood of identical semanticons for different files.

Once our system formulates a query, it is submitted to the image database for image retrieval. Despite the large size of the database, the query might be over-constrained, resulting in the service not returning an image. In this case, the query is 'relaxed' by successively removing the least frequent term. This removal process continues until the service returns an image for the query. Conversely, the context may be too specific to return an image. In this case, we split the query and treat each term as a new query, resulting in multiple sets of retrieved images. Our system, retrieves related terms by searching Lexical Freenet [Bee98], a database that indexes multiple types of semantic relationships. For example, submitting the term "grades" to Lexical Freenet, returns the related terms "school", "college", and "class".

3.3. Region Extraction

To make the images as simple and recognizable as possible, we want to remove unimportant visual information. Merely scaling a retrieved image to the icon canvas size may render it unrecognizable. It is therefore necessary for our system to detect and extract important regions in the retrieved image. We use a simple region extraction method based on image segmentation and image importance information. After segmenting the image into homogenous regions, we apply an importance map to identify important segmented regions in the image. This method tends to work well on images having few objects placed on a neutral background.

Image Segmentation: We use mean-shift image segmentation [CM02] to decompose an image into homogenous regions. The segmentation routine takes as input, the parameters: spatial radius h_s, color radius h_r, and the minimum number of pixels M that constitute a region. As with other segmentation techniques, choosing optimal parameter values is often difficult. Therefore, we over-segment the image using low values of h_r and M and merge adjacent regions based on color and intensity distributions in the CIE-Luv color space. We create a dual graph to store the segmented image regions. Nodes in the dual graph

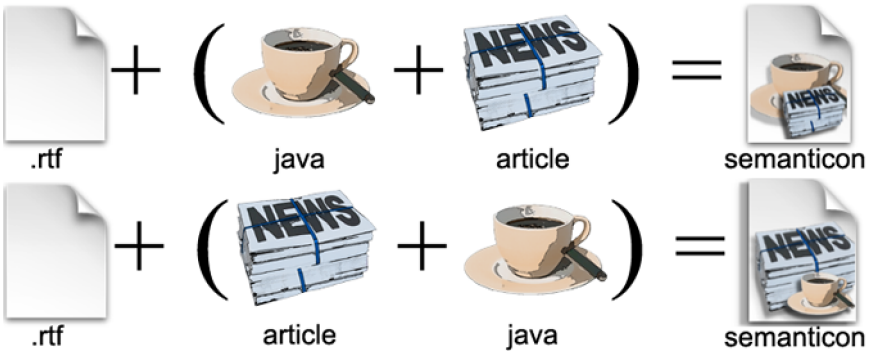

Figure 6: Two examples of automatic compositing for filenames “java_article.rtf" and “article_java.rtf" respectively.

correspond to regions in the segmented image, while edges indicate adjacency. Each node contains a color histogram of CIE-Luv components. Region merging is accomplished by combining adjacent nodes using the color similarity metric proposed by Swain and Ballard [SB91].

Importance Map: To identify important regions, we first compute an importance map that assigns a scalar value to each pixel based on a computational attention model. Like previous methods, we use measures of visual salience (e.g. image regions likely to be interesting to the low-level visual system) and high-level detectors for specific objects that are likely to be important, such as faces and signs. Our implementation computes the importance map as a scaled sum of a visual saliency algorithm [IKN98] and a face detection algorithm [RBK96]. Although we use the two similar algorithms as in Suh's work [SLBJ03], the difference is that we combine the two models rather than using them individually for extracting the most important region in the image.

The saliency and face detection algorithms take color images as input and return gray-scale images whose pixel values represent the importance of the corresponding pixel in the input image. The importance map computation can accommodate other attention models as desired. We normalize pixel values from the attention models' output images, sum them, and then re-normalize to create the importance map. We calculate an importance value for each node of the dual graph by summing the pixel values in the corresponding region of the importance map.

We extend the color similarity method of Swain and Ballard to include the additional dimension of importance and compute regions of importance by combining nodes in the dual graph. The regions of importance are formed by a clustering algorithm that considers the unexplored or unattached node with the highest importance. The clustering algorithm is applied recursively until all nodes are explored. These identified regions of importance are combined to form the extracted image.

3.4. Image Stylization

We apply cartoon coloring to the extracted image to increase the clarity of the small icons. This non-photorealistic technique helps evoke the essence of the image by omitting extraneous detail to clarify and simplify the image. Cartoon coloring is accomplished through a quantizing technique in HSV space [HE96]. The hue H of each pixel is constrained to the nearest of twelve primary and secondary colors, and the saturation S and value V are clamped to 15% and 25%, respectively.

We find that by rendering the outer contour of the extracted object with a black outline, the resulting cartoon image is aesthetically more pleasing. Our algorithm never clamps V to 0. If an object in the source image is black, the corresponding object after cartooning is gray-scale. Therefore, our black edge outlining method tends to work well even for black objects in the source images. Image stylization has the added benefit of hiding any jaggedness in the segmented images. After the image is scaled down to the size of a standard system icon, artifacts like segmentation errors tend to be further minimized. Figure 5 shows the region extraction and image stylization applied to some images.

3.5. Image Compositing

Modern computer operating systems offer a default visual style for file icons such as the Macintosh OS X icons shown in Figure 7. The generalized images are composited on to an icon template consistent with the visual style of the GUI to maintain the user's expectation of a discernable file type (e.g. default icons for a Word or HTML document). If a single query is used to retrieve images for a file's context, we create several semanticons by composing an icon template with each generalized image. If multiple queries are used, we composite the icon template with two images at a time; each image from a different query.

We composite the stylized images in the order of terms as they appear in the filename or content, starting from the left

term. We scale the first image to be as large as possible without occluding important features of the icon template (such as the blue banner on the default Word document icon, and the dog-eared page detail of the text icon shown in Figure 7). User-determined controls allow the generalized image to extend past the bounding box of the icon template. Empirically, we find that resizing the image with an additional 10% in width and height creates a good aesthetic balance, as shown in the semanticons throughout this paper. When there is a second image, it is scaled to 60% of the first image and is then composited on top of the first.

To provide us with more versatility in the compositing of images, we implement a placement algorithm that relies on the bounding boxes of two different image arrangements: side by side, and top to bottom. We choose the arrangement whose proportions best fill the available space of the default icon. Users have the ability to control the amount of overlap. In practice we found that a default of 10% width overlap with side by side compositing and about a 40% height overlap with top to bottom compositing yields serviceable semanticons. The addition of drop shadows to each generalized image helps separate images and increases the apparent depth in the resulting semanticon.

As an example, the queries for "java_article.rtf" return images including a coffee cup for "java" and a stack of newspapers for "article", as shown in Figure 6. We begin by scaling the coffee cup (which corresponds to the left term) to be as large as possible. Then, we scale the newspaper (which corresponds to the right term) and composites that image on top of the cup image. The arrangement is top-to-bottom because these image proportions best fill the template. If the order of terms differ, then the order of composing the images changes likewise.

4. Results and Discussion

The semanticons generated by our system are consistent with the visual style of contemporary GUI icons, and may convey more semantics than the file type alone, as shown in Figure 9. For example, the file "solutions/answers.xls" produces several semanticons, with images of a key, a woman thinking, a jigsaw puzzle-piece, and a light-bulb. For the file "Classes/Grades.xls" our system generates semanticons with images of a graduation cap, textbooks with an apple, a school bus, and a piece of paper with the words "A+ Good WORK!". Similarly, another file named "strategicObjectives.xml" produces a semanticon of chess pieces. "JapaneseVGA_Driver.exe" does not return any image for "Japanese VGA Driver", but composites images that depict the terms "Japanese" and "driver" with sushi and a car.

A disadvantage of using a stock photography database is that there may be certain technology specific terms or acronyms (e.g. LAN, VGA) for which the database is unlikely to have an associated image. Minor edits between versions of the same document may generate similar or identical semanticons. The distinctiveness of the filename affects the images returned by the query. Also, the system is less likely to generate identical icons with a larger, well annotated image repository. We rely on users to select distinctive semanticons from a collection of semanticons generated for each file.

Figure 7: Desktops with GUIs often have a default visual language for file icons, such as these Macintosh icons. Mac OS X Icons ©Apple Computer, Inc. and Microsoft Corporation. Used by permission. All rights reserved.

The search for images in the database is performed by matching the query terms with keywords associated with each image. Hence, the precision of the image retrieval process depends on how well the images in the database are annotated. Our region extraction method is limited by the performance of image segmentation and the importance model used for identifying important objects. If the performance of these components is insufficient, a semi-automatic version of our method can be applied where the user manually identifies the important object.

4.1. User Study

Studies have shown that search tasks in simplified environments may proceed faster than search and similar tasks executed in the full environment, and in particular, non-photorealistically rendered (NPR) images are often easier to understand than photographs [GRG04]. Based on the results of prior studies, we hypothesize that semanticons may decrease the amount of time needed to search for a specific icon among a collection of icons. This section presents the

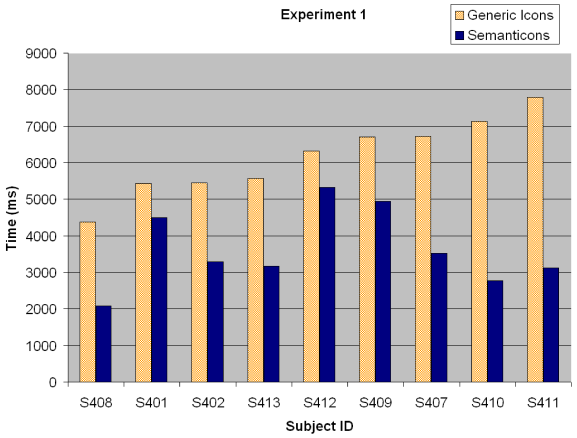

Figure 8: The average time over each of the visual search tasks is presented per participant. Results for Study 1 in which the participant performed a visual search task for generic icons, and then picked semanticons and performed visual search tasks for semanticons.

results of psychophysical studies carried out using semanticons as stimuli.

The study consists of 24 filenames, common icons, and semanticons generated with our system tested in two studies. The files include examples of HTML, Word, PowerPoint, Excel, RTF, TXT, and System icons and used icons intended to follow design guidelines for the Macintosh OS X GUI, as shown in Figure 7. The filenames are randomly chosen from a list generated from two sources: the set of recently used files as reported by the operating system on the machines of a dozen volunteers; and by searching for files of a variety of common file types using an FTP site search engine. Although a file extension is used for creating semanticons, we exclude the extension from the filename in our study. Without the extension, users are precluded from performing extension-template matching and extension hiding is often the default in several desktop GUIs such as Windows XP.

For each filename, the software system generated four semanticons. In both studies, 102 images depicting icons and semanticons are used as visual stimuli. The images are displayed on a Dell 19 inch LCD monitor at a distance of 24 inches. We set the background of the monitor to grey and each icon is presented at 128 x 128 resolution.

4.1.1. Study 1

In Study 1, there are two phases, conducted such that each participant first completes the common icon visual search, followed by the semanticon visual search. We did not counter balance our protocol as several pilot experiments showed that order did not produce a priming effect. This might be due in part to the training effect caused by the icon selection phase of our protocol. As a result, data presented in Study 1 should be considered preliminary.

Common Icon Visual Search Phase:

For each trial, participants are presented with a search filename at the top of the viewing window and a 6 x 4 grid of randomly ordered common icons. We used this size grid because pilot experiments showed that this was the minimum size with which semanticon and template icon performances showed significant statistical difference. These icons, similar to those shown in Figure 7, also include the filename text below the icon. Each participant is asked to click on the icon that matches the search filename as rapidly as possible without making a mistake. The search filename is randomly chosen without replacement for each trial. Between trials participants are presented with a gray screen for 1.5 seconds. Each of the participants performed 12 trials, including two practice trials, in this phase of the study.

Semanticon Visual Search Phase: Before the visual search for semanticons begins, each participant is presented with a filename at the top of the window and four semanticons generated by our system. The participant is asked to choose the semanticon that best represents the filename by mouse clicking on the preferred semanticon. The process is repeated for 24 file names.

Next the user performs the semanticon visual search. For each trial, participants are presented with a search filename at the top of the viewing window and a 6 x 4 grid of randomly ordered semanticons. Each semanticon includes the filename of the individual semanticon that corresponds to the semanticon the participant chose to represent the file. As a result, each participant saw a manually customized, possibly different set of semanticons. The search filename per trial is randomly chosen, from a set of 24 candidates, without replacement for each participant. Each participant is asked to click on the semanticon that matches the search filename as rapidly as possible without making a mistake. Between trials participants are presented with a gray screen for 1.5 seconds. Each of the participants performed 12 trials, including two practice trials, in this phase of the study.

In the visual search study, 9 (5 male, 4 female) graduate students, postgraduates and research staff acted as volunteers.

4.1.2. Study 1: Results

As Figure 8 shows, the semanticons are recognized on average 1.96 seconds faster than icons. For Study 1, the accuracy of recognition is 99% for the common icon (control condition) and 97% for semanticons. Using a paired t-test, we find that there is a significant statistical difference ($p < 0.001$) in performance between the timing of the judgments in Study 1. A statistical significance of 0.001 means that there is a 0.1% chance that the effects we are seeing are due to random events. It is worth noting that most psychologists agree that

such relative data comparisons are statistically significant if $p \leq .05$.

Pilot experiments demonstrated that it is important to instruct the participants to match the search filename with the icon. We noticed that participants may perform a sequential word search through the filenames in order to identify icons. We find that those participants who report using a word search for semanticons demonstrate similar speed in searches using icons and semanticons (such as Study 1 subjects S401 and S412). However, participants who use an image search technique witness a speed up in search time with the semanticons. Although the study presented here is preliminary because we did not perform a completely counterbalanced experiment, we believe these results indicate that semanticons may provide an increase in visual search speed when the semanticons themselves are used as the subject of the search. With 23 out of 32 of the filenames, participants performed better (faster search times) with semanticons than with generic icons.

4.1.3. Study 2: Memory Game

In order to assess memory retention for semanticons versus common icons, we created a task similar to the concentration or memory game. The memory game consists of a grid of randomly sorted cards placed faced down. The goal is to turn over two cards at a time. If a match is made, then the cards are removed. Otherwise, the cards are placed face down and another set of cards are turned over. The game ends when all of the pairs are found. We created a Java program similar to the card game, in which the user clicks with a mouse on a virtual card in order to have its front face revealed. We record the time it takes to make each match as well as the number of cards turned over to make a match.

Study 2 consists of two sets of trials: (1) a matching game of common Macintosh OS X icons, and (2) a matching game of semanticons. As with Study 1, the semanticon memory game was prefaced with users picking the semanticon that best represents the filename for 24 filenames. As a result, each participant saw a manually customized, possibly different set of semanticons. As in the case of the first user study, this may cause a bias in favor of our technique, and as a result, the data presented here should be considered preliminary. All of the subjects also completed a practice matching game using letters of the alphabet, before starting the trials, in order to train participants to play the game and give them time to develop a game strategy. After the practice trials, half of the participants started with the common icon matching game and the other half started with the semanticon matching game. In this study, 8 (4 male, 4 female) graduate students, postgraduates and research staff acted as volunteers.

4.1.4. Study 2: Results

Tables 1 and 2 present the total times and number of cards turned over for each participant in each game. Using a paired t-test, we find that there is a significant statistical difference in time performance between the common icon game and the semanticon game with $p = 0.018$. An analysis of the average number of cards turned over show a statistical significance for common icons versus semanticons ($p = 0.013$), with fewer cards being turned over for the semanticon memory task. There is not a statistically significant difference in performance of males versus females in the task or in the order in which the study phases are presented.

Table 1: *Average Time for Memory Games (seconds)*

Common Icons	Semanticons
112.36 s	87.98 s

Table 2: *Average Number of Cards Turned over*

Common Icons	Semanticons
70.75	59.75

4.1.5. Study 2: Discussion

The memory game study demonstrates that the semanticon memory game can be played much faster than the common icon memory game. In addition, the memory game with semanticons requires fewer cards to be turned over, possibly indicating that it is much easier to remember previously revealed semanticons. From this we conclude that semanticons may aid viewers in short-term memory intensive tasks. It is possible that our experiment may have introduced a bias because the users pick semanticons before performing the matching task and there is not a similar task with the common icons. Further studies should be conducted to remove this potential bias as well as to examine whether semanticons aid in tasks requiring long-term memory.

5. Conclusion

"Visual puns combine two or more symbols (picture and/or text) to form a new meaning. The viewer must mentally elaborate on the visual stimulus to interpret this metaphorical relationship" [Abe94]. We speculate that some of the success of the semanticons may be due to the formation of this type of visual pun or metaphor in the mind of the viewer.

This paper documents a technique that facilitates visual communication by automatically creating GUI icons. These icons, called *semanticons*, are created using images that are semantically linked to the files' name, path and contents. The algorithm can be completely automated and produces multiple potential icons for a given file.

We report the results of two psychophysical studies carried out using semanticons as stimulus. Results show that semanticons may yield quicker response times in visual search tasks and improved retention in short term memory tasks.

6. Acknowledgements

We thank Jack Tumblin, Rachel Gold, Bryan Pardo, Amod Setlur and the anonymous reviewers for all their help and feedback in improving the content of the paper. Thanks to Marcia Grabowecky for reviewing the user studies.This material is based upon work supported by the National Science Foundation under Grant No. 0415083. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[Abe94] ABED F.: Visual puns as interactive illustrations - their effects on recognition memory. In *Metaphor and Symbolic Activity* (1994), vol. 19, pp. 45–60. 8

[AL98] ANQUETIL N., LETHBRIDGE T.: Extracting concepts from file names: a new file clustering criterion. In *ICSE '98: Proceedings of the 20th international conference on Software engineering* (1998), IEEE Computer Society, pp. 84–93. 3

[Bee98] BEEFERMAN D.: Lexical discovery with an enriched semantic network. In *In Proceedings of the Workshop on Applications of WordNet in Natural Language Processing Systems* (1998), ACL/COLING, pp. 135–141. 4

[Byr93] BYRNE M. D.: Using icons to find documents: simplicity is critical. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems* (1993), ACM Press, pp. 446–453. 1

[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (2002), 603–619. 4

[FFH01] FOGARTY J., FORLIZZI J., HUDSON S. E.: Aesthetic information collages: Generating decorative displays that contain information. In *UIST 2001: Proceedings of the 16th annual ACM symposium on User Interface Software and Technology* (2001), ACM Press, pp. 141–150. 4

[GRG04] GOOCH B., REINHARD E., GOOCH A.: Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph. 23*, 1 (2004), 27–44. 6

[HE96] HEALEY C. G., ENNS J. T.: *A Perceptual Colour Segmentation Algorithm*. Tech. rep., 1996. 5

[HH90] HENRY T. R., HUDSON S. E.: Multidimensional icons. *ACM Trans. Graph. 9*, 1 (1990), 133–137. 1

[IKN98] ITTI L., KOCH C., NIEBUR E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 11 (1998), 1254–1259. 5

[Ind] http://istockphoto.com/. Image stock photography. 4

[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. 1

[Moy94] MOYES J.: When users do and don't rely on icon shape. In *CHI '94: Conference companion on Human factors in computing systems* (1994), ACM Press, pp. 283–284. 1

[RBK96] ROWLEY H. A., BALUJA S., KANADE T.: Human face detection in visual scenes. In *Advances in Neural Information Processing Systems* (1996), Touretzky D. S., Mozer M. C.,, Hasselmo M. E., (Eds.), vol. 8, The MIT Press, pp. 875–881. 5

[RD03] RUSSELL D. M., DIEBERGER A.: Synthesizing evocative imagery through design patterns. In *Hawaii International Conference on Systems Science* (2003). 1

[RK88] RUBENS P., KRULL R.: Communicating with icons as computer commands. In *SIGDOC '88: Proceedings of the 6th annual international conference on Systems documentation* (1988), ACM Press, pp. 25–34. 2

[SB91] SWAIN M., BALLARD D.: Color indexing. *International Journal on Computer Vision 7*, 1 (1991), 11–32. 5

[SLBJ03] SUH B., LING H., BEDERSON B. B., JACOBS D. W.: Automatic thumbnail cropping and its effectiveness. In *UIST 2003: Proceedings of the 16th annual ACM symposium on User Interface Software and Technology* (2003), ACM Press, pp. 95–104. 1, 5

[SWY75] SALTON G., WONG A., YANG C. S.: A vector space model for automatic indexing. *Commun. ACM 18*, 11 (1975), 613–620. 3

[WFR*01] WOODRUFF A., FAULRING A., ROSENHOLTZ R., MORRSION J., PIROLLI P.: Using thumbnails to search the web. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), ACM Press, pp. 198–205. 1

Figure 9: Examples of semanticons generated by our system.