Kineticons: Using Iconographic Motion in Graphical User Interface Design

Chris Harrison[1] **Gary Hsieh**[2] **Karl D. D. Willis**[3] **Jodi Forlizzi**[1] **Scott E. Hudson**[1]

[1]Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213
{chris.harrison, forlizzi, scott.hudson}
@cs.cmu.edu

[2]College of Communication
Arts and Sciences
Michigan State University
East Lansing, MI
garyh@msu.edu

[3]Computational Design Lab
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
kddw@cmu.edu

ABSTRACT

Icons in graphical user interfaces convey information in a mostly universal fashion that allows users to immediately interact with new applications, systems and devices. In this paper, we define *Kineticons* - an iconographic scheme based on motion. By motion, we mean geometric manipulations applied to a graphical element over time (e.g., scale, rotation, deformation). In contrast to static graphical icons and icons with animated graphics, kineticons do not alter the visual content or "pixel-space" of an element. Although kineticons are not new – indeed, they are seen in several popular systems – we formalize their scope and utility. One powerful quality is their ability to be applied to GUI elements of varying size and shape – from a something as small as a close button, to something as large as dialog box or even the entire desktop. This allows a suite of system-wide kinetic behaviors to be reused for a variety of uses. Part of our contribution is an initial kineticon vocabulary, which we evaluated in a 200 participant study. We conclude with discussion of our results and design recommendations.

ACM Classification: H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

General terms: Human Factors

Keywords: Icons, menus, windows, drag, kinetic, animation, dynamic, design, GUI, look and feel, affordances.

INTRODUCTION

Graphical icons, without doubt, are a critical component of computing's successes. Their ability to convey information in a fairly universal fashion allows users to immediately interact with new applications, systems and devices [16,33]. People's ability to pick up e.g., an iPhone for the first time and successfully use a wide variety of its features is a remarkable achievement. Conversely, walking up to a command line interface offers no similar affordances. The user, even if knowledgeable with other command line systems, must essentially follow a trial-and-error approach.

The content of *graphical icons* is typically pictographic (pictorial resemblance to a physical object or action) or ideographic (a visual representation of a concept). However, their visual properties can also be modified along other dimensions to convey extra information. For example, a common metaphor for an inaccessible or hidden item is a ghosted, faded or grayed-out appearance. This modification is iconographic in and of itself, and can be applied to a wide variety of "icons" (i.e., GUI elements), including files, applications, dialog boxes, title bars, drop down menus, menu items, buttons, cursors, sliders, and many more.

Highly intertwined with graphical icons is *animation*. One form of animation is changes to graphical content of an icon over time [2,3,10,14,42]. A progress bar [37] is a good exemplar – its visual state changes over time to convey the progress of an extended operation. Buttons could also depict their action with an animated sequence – a draw tool might display a small pen drawing a line, as seen in [3,10]. Many other interactions exist: an item that has become accessible could smoothly fade in [35] or highlight [32]; an afterglow could be used to convey the recentness of an interface change [6]; files that are dragged could feature a graphical motion blur [13]. The temporal staging of these animations can also be used to convey information [18].

Several non-visual iconographic forms exist, which are also powerful. Computing devices can often output sound, allowing for *auditory icons* [15,19,20]. These typically depict audio relating to real world events, objects or actions to convey meaning. For example, a ticker sound might indicate a news item has arrived; a cheer might be used for a sports-related item. Related are *earcons* [8,9,12,22], which are synthesized sounds that are easy to learn and associate meaning with. *Hapticons*, as the name suggests, convey iconic information haptically [17] (see also tactons [11]). Although almost exclusively done though vibro-tactile means at present, texture could also be used [23].

In this paper, we define a new type of iconographic scheme for graphical user interfaces based on motion. We refer to these "kinetic icons" as kineticons. A kineticon is the result of a sequence of geometric manipulations (e.g., scale, rotation, deformation, translation) applied to pre-defined regions of a graphical element over time. We call these ma-

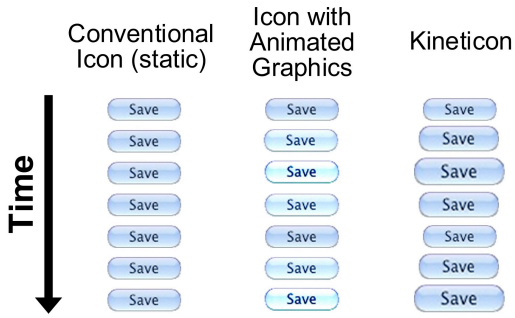

Figure 1. The appearance of a conventional icon, an icon with animated graphics, and a kineticon over time. A MacOSX save button serves as the example icon.

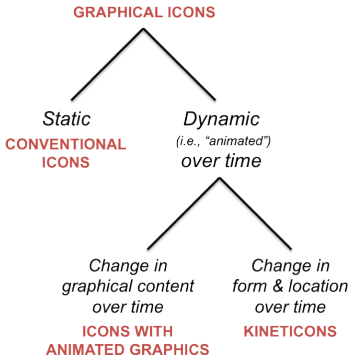

Figure 2. A taxonomic breakdown of graphical icons.

nipulations *kinetic behaviors*. The bounds of these regions (potentially one encompassing the entire element) could be defined in several ways (e.g., a 20x20 pixel region in the top left, or the top 50% of the item - similar to e.g., how CSS can define absolute or relative sizes). This content independence allows the kinetic behaviors to be applied to almost all GUI elements, including those with radically different shapes and sizes.

In contrast to static icons and icons with animated graphics, kinetic behaviors do not alter the visual content or pixels' RGBA values (i.e., "pixel-space") of an element (where as e.g., fades, blurs, tints, and addition of new graphics exclusively changes the pixel-space). Stated differently, pixels in an icon can be moved, rotated, stretched, and so on - but not altered or added to. Figure 1 illustrates the difference between a standard graphical icon, an animated icon and a kineticon. A taxonomic breakdown is provided in Figure 2.

The contribution of this work is multi-fold. Foremost, we define kineticons and clearly delineate their scope. This offers a name to an existing, but loosely applied interaction technique. This process involves carefully teasing out the difference between *icons with animated graphics* and *kineticons* (typically both referred to as "animated icons"). This distinction should help to better classify and target future research efforts. We also briefly review existing in-the-wild kineticons, before outlining the many properties of motion that make it an excellent dimension to leverage for interface design. Additionally, we review several popular sources of inspiration kineticon designs can draw upon. To illustrate and motivate our approach, we introduce eight example classes of interactions where kineticons could enhance current GUIs. We share our initial kinetic behavior vocabulary, and the results from our 200 participant user study, from which we draw many conclusions and design recommendations. Finally, throughout this process, we coalesce a wide variety of relevant work to aid future researchers.

EXISTING KINETICONS

Importantly, although we present a definition of the space, kineticons are not new. Although much previous academic work has been grossly labeled "animated", most would be considered "icons with animated graphics" – a handful would be classified as kineticons [13,29,30,43,44]. Animation in graphical user interfaces dates back more than two decades, with Baecker et al. [3] and Bodner et al. [10] pioneering its first uses.

Beyond academic work, several successful kineticons exist in production systems used today. One of the oldest and most successful kinetic behaviors is window and menu transitions – in particular, those that slide or expand the window in from a hidden or minimized state. This feature debuted as early as Windows 95, and traces its origins back to at least the first Macintosh interface (where a series of ever-larger gray rectangle outlines was used to graphically animate the opening). Overall, the most common use of kineticons at present is to aid in transitions between states, which helps to provide context and causality [14,21,41]. However, as we will see, kineticons have many more uses.

Apple's Aqua interface contains some of the most visible kineticons. For example, when an application is launching, its icon "bounces" in the dock (Figure 5, *Bounce*). A slightly different bounce effect is used when an application requires user attention (e.g., printer out of paper). Window minimization and maximization can use either a "scale" or "genie" kineticon (Figure 5, *Suck In*). Finally, if an incorrect password is entered into the login screen, the window shakes left to right (Figure 5, *Shake No*).

The latest generation of touch-driven interfaces tends to feature very rich graphics, animation and kineticons. The iPhone, for instance, employs a small, but powerful set of kineticons. Of particular note is the "wobbly" state icons assume if they are able to be dragged (Figure 5, *Rumble*). Also effective is the "zooming" effect used to transition from the home screen to an application (Figure 5, *Zoom*).

WHY MOTION?

The benefits of using motion are many and diverse. Chief among them is the ability to readily apply motion to GUI elements of different sizes. For example, a simple "wave" kineticon could be applied to something as small as a close button or file icon, to something as large as window of files or even the entire desktop. This allows a suite of system-wide kinetic behaviors to be reused for a variety of purposes. Importantly, this allows the iconic set to be smaller,

since behaviors unique to toolbar items, windows menus, application icon, etc, are not necessary. It also establishes a common iconography that can be used throughout the system, potentially increasing ease of use. For example, a movable file, a movable window and a movable menu item can all use the same kinetic behavior.

Systems wishing to offer a similar suite of reusable graphical-level manipulations have two primary options: 1) provide a series of sub-icons or badges that could be added to existing icons, or 2) provide a series of whole-icon graphical manipulations. This works fairly well in practice with simple augmentations, such as superimposing a number badge or graying out an icon. Indeed, graying out a GUI element is perhaps the most widely used visual metaphor; however, the design space is very small: one state. Additionally, both of these schemes graphically change the content inside elements, which has the potential to alter (even destroy) the appearance and affordances of a GUI item.

In general, the graphic design of icons and interactive elements lies in the realm of the interface and application designer; the system typically does not know the application-level state or appearance of an element. For example, the superimposition of a "printing badge" onto an application icon might cover an application-level badge (e.g., number of buddies online). Adding additional complexity is the fact icons can be designed in numerous styles, including flat, isometric, cartoon, and photo-realistic. Superimposing a generic e.g., "printing badge" could harm the look-and-feel of an application.

Kineticons suffer less from these complications. In general, they can be applied to almost all graphical styles and operate in harmony with other dynamic visual elements. It is entirely possible to have an icon with animated content that is grayed out, and also performing a kinetic behavior. Indeed, motion can act as an additional dimension to all aforementioned iconographic schemes, adding more expressive power to interfaces. However, similar to other iconographic schemes, combining several kinetic behaviors is likely to be problematic.

Motion also has considerable psychophysical benefits. Foremost, we possess an innate understanding of how the world operates physically. Metaphors such as mass, rigidity, and inertia are readily portable to digital domains and make operations more intuitive and expressive [24]. Moreover, humans have incredibly parallelism in their visual processing and for motion in particular [5,27,47]. Not only can multiple motions be viewed and interpreted at any given instant [40], but we can perform complex grouping tasks (e.g., common fate principle in Gestalt psychology) [31]. Although humans have some ability to ignore or attend to motion, it is widely accepted that motion in interface design is attention grabbing. Thus, motion must be judiciously applied so as to minimize unnecessary distractions.

Equally impressive is our ability to perceive motion in our peripheral vision, which operates at a far lower resolution than our fovea, but has high sensitivity to motion. This potentially allows kineticons to be detected and identified with peripheral vision. This stands in contrast to graphical changes, such as overlaying the number of unread messages onto the email application icon (as seen in MacOSX). Although the eyes could detect a change has occurred, it would require a glance to read the number. Although we can only speculate, it seems likely kineticons could have implications to users with impaired vision, which typically impacts visual acuity, but not motion perception.

DESIGN

At present, GUI design is typically motivated by the iterative design process of interface designers. Researchers continue to develop ways to systematically apply design and perception knowledge to aid system builders. As an initial step in our research, we considered six popular sources for design inspiration, which we share below. Importantly, these sources can heavily leverage our innate perpetual abilities and knowledge of the world. In the same way that good graphical icons are often pictographic, kineticons can be emblematic of real-world motions.

Biological Motion

Scientists in many disciplines have researched animal and human body motion. From the ancient Greeks to DaVinci, from Muybridge to Disney, researchers in many disciplines have been fascinated with how humans stand, move, and gesture. Today, studies of motion are conducted in robotics, graphics, and biomechanics and benefit such disparate application areas as athletics, rehabilitation, ergonomics, animation, virtual training, and humanoid robotics.

The utility of employing animal and human motion is clear; even eight-month-old infants recognize and fixate longer on causal human motion than non-causal motion [28]. Recognition and understanding of human motion is core to many basic human skills, allowing us to correctly interpret non-verbal communication, to recognize friends from a significant distance, and to acquire new skills through imitation.

The most dramatic demonstration of our perceptual abilities is with light point figures. In these experiments, an actor is augmented with small lights on his shoulders, elbows, knees, ankles, and wrists. When the actor sits motionless, viewers perceive nothing more than a random constellation of disconnected elements. However, when the actor moves, viewers immediately perceive human motion, even though the actor himself is invisible [25,38]. Researchers have subsequently discovered that people are able to make quite sophisticated judgments about the motion, for example, deducing the gender of the actor [4].

Gestures

Gestures are a class of motion that is explicitly designed to convey information (where as locomotion or posture is more suggestive or causal). Robust meanings can be expressed via simple gestures such as a head nod, shoulder shrug, thumbs up, or rapid foot tapping [26,36]. For example, a single hand can either beckon or ward away and, by changing motion parameters in the gesture, the urgency of the message can also be made apparent.

Figure 3. Illustration of how MacOSX indicates drag-drop applicability (Microsoft Windows has a similar behavior). A) The user begins to drag a JPG document. B) User drags the file over the QuickTime Player application. There is no highlight because QuickTime is not able to open JPG files. C) User drags the file over the Preview application. This application can open JPG files; the blue highlight indicates the file can be "dropped" here.

Human gestures make for excellent kineticons, primarily because of our extensive training with them. Actions such as wave, shake head (i.e., no), stretching (e.g., after getting out of bed), and jumping up and down with hands flailing have immediate kinetic translations to icons. Although less visible to us, animals signal to each other in equally sophisticated ways. Gestures most familiar to us are from domesticated species, such as a dog's tail wagging.

Organic Motion

Beyond animal locomotion and gestures, there is a plethora of smaller motions inspired by organisms and organic processes, to which we are intimately familiar. For example, the blossoming of a flower, the beating of a heart, and pupils dilating. These iconic actions in nature can have very strong and particular associations that can be readily leveraged in interface design (e.g., a flower opening has strong connotations of rebirth and freshness).

Mechanical Motion

Today's GUIs have been imitating mechanical motion successfully for several decades: e.g., toggles, knobs, buttons and sliders. Designers have codified this knowledge into a set of guidelines that can be used reliably for interface design [7,24]. Furthermore, our brains have dedicated areas for processing non-biological-motions, including those of artificial or mechanical origin [38,39]. The huge number of designed artifacts in the world gives us a rich library of mechanical motions from which to draw upon. For example, the "tick-tock" of a clock hand, the unzipping of a garment, and the cyclical motion of an engine. Even now-obsolete representations, such as the motion of a steam locomotive's pistons, continue to have strong connotations.

Physics and Natural Effects

Conventional GUIs tend to rely solely on digital artifacts. The integration of a physics engine into these experiences has been shown to be powerful [1,24,46]. More importantly, it can leverage our innate perceptual abilities and extensive training with physical objects and how they behave in the real world [38,39]. We can simultaneously leverage this knowledge by borrowing motions from natural and physical effects. For example, how a leaf falls from a tree, glass shatters, paper folds, a sign blows in the wind, a swing rocks back and forth, or liquid drains from a sink.

Cartoon Conventions

Finally, cartoons and comics provide an interesting source of artistic embodiments of motion [34,45], which are typically caricatures of real-world motions. This exaggeration translates well to animation and certainly to the 2D desktop of conventional graphical user interfaces [13,29,43,44]. The canonical example is that of "squash and stretch" - defining the rigidity and mass of an object by distorting its shape during an action [13,29]. For example, when a ball hits the ground, its shape is flattened vertically. Upon leaving the surface, it squeezes horizontally, stretching upwards, as if leaping from the surface.

The fanciful content of cartoons led animators to design new motion metaphors [45], which have entered pop culture. For example, the rapid spinning of The Road Runner's legs or Superman changing outfits. Although often supplemented with motion blurs and graphical "streaks", the action can often be conveyed effectively using motion alone.

EXAMPLE USES

We now introduce and discuss a small subset of possible uses for kineticons in modern graphical user interfaces. We focus on three classes of interactions: the state of a GUI element, state changes, and finally, positive/negative events (e.g., an operation succeeded or failed). This initial set is by no means comprehensive; instead mostly illustrative.

State

There are many classes of state that a digital item can assume and are important in user interfaces; we touch on four:

Ability/Affordance of an item

Kinetic behaviors could be used to convey the abilities or affordances of a graphical element. For example, the ability to be moved (e.g., dragged with the mouse) could apply to a variety of elements, including file icons, toolbar items, menus, menu bar items, windows, and perhaps even the whole desktop (if in a multi-screen environment). A single kinetic behavior conveying "is movable" could be applied to this diverse set (the iPhone uses a rumbling kineticon to convey this affordance; Figure 5, *Rumble*). Other states might be whether an item is delete-able, edit-able, open-able, and resizable, to name a few.

One GUI operation that is particularly awkward at present is drag and drop of files onto applications. In many systems, whether the application is a possible drag-drop destination for a file is not known until the dragged object is over the application icon. If the operation is possible, the application icon highlights; if not possible, the icon is unchanged. Figure 3 illustrates this behavior. The key problem with this interaction is one has to scan over application icons in order to find an applicable opener.

Kineticons offer an immediate solution to this. When dragging a GUI element, interface elements that applicable drop-targets could exhibit a kinetic behavior with a receptive connotation. This could not only apply to application icons, but also disks, application windows (e.g., photo editor or email), and file browsers (Figure 4). Interface elements with inapplicable state can either remain motionless, or exhibit a not applicable kinetic behavior. This would allow users to effortlessly identify possible drop-targets.

Needs Attention

An important state in interfaces is one that requires the users attention to proceed or resolve. This might occur if a location needs to be specified for a file save, the printer is out of paper, a laptop is running low on energy, a file has finished downloading, and countless other states. It need not convey a positive or negative connotation, just one of urgency. Kinetic behaviors could be used to signal users. Indeed, MacOSX uses an iconic "double bounce" kinetic behavior for this purpose - although it is limited to icons in the dock. Again, as mentioned many times previously, a "needs attention" kinetic behavior could also be applied to a save dialog, a print button, a battery indicator, and so on.

Operation is Progressing

Another common state for GUI elements to occupy is one of an operation in progress. This can occur when a file is downloading, a document is spooling to a printer, and a new application is being installed. In many cases, the duration of the processes can be computed (e.g., 10Mb file downloading at ~100kbps takes roughly 100 seconds). Kineticons could reflect the progress or speed of an operation through motion in the same way progress bars depicts it graphically. Alternatively, a progress bar could be used to convey percent-done [37], while motion is used to indicate e.g., download speed. This could allow for more ready comparison between processes and an overall greater awareness of state.

Launching and Opening

Highly related to the previous class of "operation is progressing" is launching and opening of elements. There are two key differences. First, launching and opening often has indeterminate duration. Second, it has a slightly different connotation: it is the start of a new process, with something new appearing or becoming available when launching/opening is complete. For example, a "connecting" kinetic behavior might be applied to a server share being mounted, where as a "progressing" kinetic behavior might be used when a file is being downloaded from the server.

State Changes and Events

In addition to state, such as a document being editable, there are state changes and events that could be facilitated by kineticons. We now briefly discuss four classes:

Entrance

This class can be thought of as the birth or arrival of a new item. In other words, from something not existing to existing. Instead of simply instantly appearing, the appearance of a new item can eased though kinetic behaviors. Examples include the mounting of a USB drive on the desktop, the appearance of a new application following an install, the opening of a new file browser.

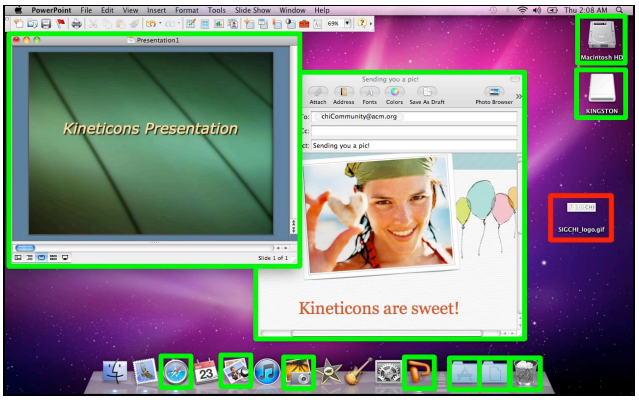

Figure 4. A user dragging a JPG file (red highlight) has numerous potential drop-targets (green highlight), including applications, the trash, folders, disks, a PowerPoint presentation and an email. These regions could be indicated to the user through subtle kinetic behaviors.

Departure

This class of state change is the opposite of entrance – it is an item that transitions from existing to not existing. This might be a DVD ejecting from the desktop, a file being deleted, a dialog box closing, and a buddy logging out of an instant messenger application.

Replace/Update

This special case is two actions occurring in parallel: the end of one state and the beginning of another. As such, the transition between the two can be represented by two kinetic behaviors: *departure* of the old element, and *entrance* of the new one. Examples include the updating of an application to a new version, the replacing of one file with another, or advancing to the next slide in a presentation. Also, options available in a toolbar could update given the type of object selected in the interface.

Positive/Negative Events

The outcome of many operations and actions in user interfaces are binary – the install either succeeded or failed, the printer is available or not available, the disk had space for the transfer or did not have space. Kineticons are well suited to conveying the outcome of an event, either user-driven or application-driven [14].

MacOSX offers a paradigm example. If a password is accepted, the dialog disappears. However, if the password is rejected, the dialog shakes left to right briefly, essentially a "no" gesture (Figure 5, *Shake No)*. This behavior is highly effective – there is no reason why this kinetic behavior could not be applied to interface elements beyond a dialog box. For example, entering letters into a numerical telephone text field (interactor), attempting to open an Excel file in Adobe Photoshop (application icon), or attempting to operate an out-of-order ATM (whole screen).

Figure 5. Our proof-of-concept kineticon vocabulary.

Figure 6. Kineticon applied to an application-style icon, menu item, dialog box, and entire desktop.

Gender	%	Age	%	Salary	%	Daily Comp. Use	%
Male	33	18-24	27	<$20k	35	<1 hr	2
Female	67	25-30	27	$20k-30k	17	2-3 hr	25
		31-40	27	$30k-$50k	24	4-6 hr	37
		41-50	10	$50k-$75k	16	6-10 hr	21
		51+	9	$75k+	8	> 10 hr	15

Table 1. The demographics of our MTurk Participants

PROOF-OF-CONCEPT KINETICON VOCABULARY

To motivate, illustrate and evaluate kineticons, we created a set of 39 kinetic behaviors. Obviously the set of possible behaviors and GUI elements to apply them to is limitless. To cover the design space with a fairly diverse set of kinetic behaviors, we created between 4 and 7 iconic motions for an interaction from each class we discussed in our Example Uses section. In some cases, we designed very specific behaviors (e.g., "able to be moved"); in other cases, very general ones (e.g., "progressing"). We also selected interactions that were conceptually similar (e.g., "progressing" and "loading") to see if well-designed kinetic behaviors could convey subtle differences. Interactions that were highly distinctive (e.g., "departure") were also included.

To aid our design process, we drew on the sources of design inspiration noted earlier. We also included five kinetic behaviors found in MacOSX and the iPhone, primarily for later discussion. Figure 5 provides an overview of our initial kineticon vocabulary. Motion is hard to convey on paper, so please also refer to the Video Figure.

USER STUDY

We conducted a user study to answer two key, high-level questions. Foremost, can motions be iconic, and recognized across users? This is a fundamental property kineticons must exhibit if they are to be useful in the design of graphical user interfaces. Our second question was whether or not kinetic behaviors can be reliably applied across different GUI elements, a key property we champion in this paper.

There are several important secondary outcomes from our study. Evaluating our proof-of-concept kineticons allows us to suggest an initial vocabulary of kinetic behaviors interface designers can employ. It also allows us to gauge how well our design intention matched the interpretations of our participants. Concurrent with this is the evaluation of current in-the-wild examples, most notably those used in Apple products. We conclude with high-level design recommendations based on our interpretations of what worked and what did not.

Setup & Measures

We applied our 39 kinetic behaviors to a generic application-style icon (Figure 5). These were rendered as movies for the purposes of the study, but are fully programmatic. We then showed this set of 39 kineticons to a three independent coders, whom (through a consensus vote) selected the best kineticon in each of our eight interaction examples. We then chose four from these (*Heart Beat*, *X-cross*, *Running and Blowing in the Wind 2D*) to be applied to three other GUI elements beyond an application icon: a menu item, a dialog window, and entire desktop (Figure 6 and Video Figure). This added 12 additional kineticon videos to our test set (51 items total).

In order to recruit a large and diverse set of participants, important for making claims of broad recognition, we conducted an online study through Amazon's Mechanical Turk (MTurk). Remote participants watched videos of the kineticons, one at a time. These videos were presented on our study's webpage through an embedded video player, which automatically started and looped.

While watching the videos, participants were asked to rate how strongly they agreed or disagreed (five-point Likert scale) with the following eight interpretations of the motions seen in the video.

1. Item is progressing with an operation.
2. Item has been replaced with a new item.
3. Item cannot complete the user's request.
4. Item is newly available.
5. Item is no longer available.
6. Item needs attention.
7. Item is opening or loading.
8. Item is able to be moved.

Each participant rated a total of ten kineticons, drawn randomly from our pool of 51 videos (the ordering of the interpretations was also randomized per participant). After completing ten ratings, participants answered a set of background questions.

Participants

Participation in this study was limited to participants residing in the US, have previously submitted at least one unit of work (MTurk HIT) and have approval ratings (previously approved tasks / all submitted tasks) greater than 80%. This pre-screening was to ensure both basic English proficiency and to minimize the number of low-quality MTurk workers. 200 MTurk workers participated in the study (67% female), who were paid $2. Most of our participants spend between 2-6 hours on their computers per day (62%). Table 1 provides a breakdown of demographics.

Statistical Analysis

To control for individual biases, participants' Likert ratings for each interpretations were standardized (into z-scores). To do this, each individual's rating had their mean rating (for that interpretation) subtracted, and was then divided by the standard deviation of their ratings (for the interpretation). When standardized, 0 means an average rating for that interpretation, +1 means a rating that is one standard deviation above the average, and -1 means a rating that is one standard deviation below the average. Ratings from twelve

participants were excluded in our analyses due to zero variance (e.g., all 1s) or spending too little time on ratings (under 10 seconds).

To calculate which kineticons were rated highest for each interpretation, an ANOVA was used. The eight interpretation ratings were used as dependent variables and kineticon type (39 possibilities) was the independent variable. Tukey's HSD was then used in our post-hoc analyses to test the differences between kineticons.

To determine if kinetic behaviors were interpreted the same way across different GUI elements, we used a regression model. Interpretation ratings for the four sets (*Running*, *Blowing in the Wind 2D*, *Heart Beat*, *X-cross*) were analyzed, one set at a time. In these analyses, the interpretation rating was the dependent variable and the element type (application icon, menu item, dialog, desktop) and interpretation (across the 8 possible interpretations) were the independent variables. Tukey's HSD was also used here to test the differences between interpretations.

RESULTS AND DISCUSSION

We conducted several analyses to investigate different performance aspects of kineticons. We interweave this review of the results with discussion and design recommendations where appropriate. We first discuss the 39 application kineticons (Figure 5), and conclude the section with discussion regarding applicability across GUI elements.

Performance of Initial Kineticon Vocabulary

We conducted oneway ANOVA analyses on each item and found that 36 of the 39 kineticons are significantly more likely to convey some meanings than others (all at $p \leq 0.01$, except for *X-cross*, *Wave*, and *Circular Wave*). The strengths of Kineticons in a given interpretation were generally evenly distributed. To group the best performing kineticons, we define two thresholds, $\mu+\sigma$ ("best") and $\mu+2\sigma$ ("top"), based on the distribution of standardized z-scores. Additional statistically significant results are noted inline. We consider possible interpretations of the results if themes are present in the successful kineticons.

"Item is able to be dragged by the mouse"

Six kineticons operate above the best threshold: *Hanging Sign*, *Blowing in the Wind 2D*, *Blowing in the Wind 1D*, *Jump Wave*, *Jump Reach and Running*. The first three were designed to convey the ability to be moved, suggesting our designs were successful in generating the right interpretation. Moreover, these three kinetic behaviors perform strongly in only this category, suggesting a primary interpretation, which is the desirable outcome.

The high performance of the other three kineticons is curious. Unfortunately, *Jump Wave* has an additional strong interpretation of "needs attention", diminishing its benefit and utility (i.e., leading to confusion). *Running* and *Jump Reach*, however, do not have strong alternative interpretations, and join the first three as a strong indicators of "is draggable by the mouse". It is possible the reaching and running anthropomorphizations suggested to participants the icon "wanted to go somewhere", and thus was movable. A waving gesture, on the other hand, has less of a goal- or destination-oriented connotation.

"Item needs attention"

We hypothesized that most kineticons would rank highly in the "needs attention" interpretation due to motion's inherently attention-grabbing quality. However, 20 of the 39 kineticons had negative z-scores, suggesting that there is actually a spectrum of "attention-grabbing-ness". Seven kineticons yielded rankings above the best threshold: *Shake No*, *Jump Wave*, *Cartoon Bounce, Rumble*, *Whole-Icon Wave*, *Heart Beat* and *Alarm Clock Ringing*. The latter four perform well in this interpretation alone, suggesting they could be good kineticons to employ.

The two waving-inspired kineticons clearly leverage the human gesture, which typically has an expectation of attention. We intended both of these kinetic behaviors to elicit this effect, and it appears to have succeeded. *Rumble*, *Shake No*, and *Alarm Clock Ringing* have a similar vigorously shaking appearance – embodying a lot of energy and urgency. This appears to have been translated to a need for attention (the similar *Earthquake* kineticon lies just under the best threshold). The reasons for the success of *Cartoon Bounce* and *Heart Beat* are not immediately obvious.

"Item is progressing with an operation"

The *Steam Engine* kineticon is the strongest in this interpretation, surpassing the top threshold by a wide margin. Unfortunately, it also performs well in the conceptually similar "loading" interpretation. Five others exceed the best threshold: *Spin*, *Drain*, *Cartoon Bounce*, *Pivoting*, *Rocking Chair*.

The commonality between *Steam Engine*, *Spin*, *Pivoting* and *Rocking Chair* is their highly cyclical and mechanical nature, like an engine running. Moreover, these represent four of the six kineticons that exhibit this behavior - the other being *Hanging Sign and Whole-Icon Wave,* neither of which exude a strong feeling of mechanical motion.

"Item is opening or loading"

Seven kineticons perform above the best threshold. Four are kineticons we intended to convey "entrance": *Fold In*, *Inflate*, *Spin In*, and *Zoom*. Clearly our choice of the words "opening" and "loading" provided the impression of something new being instantiated, and thus appearing. Interestingly, despite a subtle, if not ambiguous difference from the "progressing with an operation" interpretation, there were only two overlapping best-performing kineticons: *Steam Engine* and *Spin*. Finally, *Frantic*, a more energetic version of *Running*, was also highly ranked.

"Item is newly available"

This interpretation was more polarizing than others, with kineticons tending to either be in strong agreement or disagreement. Consequently, ten kineticons fall above the best threshold. All five of our intended "entrance" kineticons are included in this set (*Spin In*, *Inflate*, Zoom, *Thrown In*, and *Unfold*), showing the metaphor of appearing from nothing is unsurprisingly a strong signal new availability. All four of

our conceptually related "Replace/Update" kineticons also operate above the best threshold (*Fall Over Reveal*, *Superman*, *Shatter Reveal*, and *Unzip*). Although unintended, this result actually suggests the metaphors we used to signal the arrival of a new item are strong. Additionally, the *Bounce* kineticon performed well for unknown reasons.

"Item is no longer available"

This interpretation had strong results, with three kineticons in the top tier and three above the best threshold. The clear winners were the intended "departure" kineticons: *Shatter*, *Suck In*, *Drain*, and *Fall Over (*Tukey's HSD; $p<.05$ compared to all other kineticons, but not compared to each other). As noted subsequently, departure is viewed as similar to "cannot complete" interpretation, and consequently, *Shatter*, *Fall Over* and *Suck In* perform well in that category. *Drain*, as noted previously, is interpreted as a "progressing" kineticon. The remaining two high performing kineticons are both "Replace/Update" kineticons: *Fall Over Reveal* and *Unzip* – both of which feature a departing action. Overall, it is apparent the kineticons designs are working. However, some refinement might be needed to better separate them from the clearly different "cannot complete" interpretation.

"Item has been replaced with a new item"

Participants had very decisive reaction to this interpretation. Four kineticons operate in above in excess of the top threshold (none scored between best and top thresholds). Unsurprisingly, given the distinctive sequence of these kinetic behavior sequences (two kineticons in parallel), *Shatter Reveal*, *Fall Over Reveal*, *Superman* and *Unzip* work well ($p<.05$ compared to all other kineticons, but not among the four).

"Item cannot complete the user's request"

The best performing kineticon was *Shatter* (top tier). It is clear this metaphor has a strong "breaking" and negative connotation. Nine other kineticons performed above the best threshold: *Head Side-to-Side*, *Pulse*, *Fall Over*, *Alarm Clock Ringing*, *Flapping*, *Suck Out*, *Earthquake*, and *Rumble*.

Four of the five kinetic behaviors we designed with a "cannot complete" connotation were highly ranked. Interestingly, many of the behaviors intended to convey departure also ranked highly. It appears the two are linked conceptually – loosely, something that fails typically ends or disappears.

Performance of Existing Kineticons

We incorporated five kinetic behaviors found in MacOSX and the iPhone into our proof-of-concept set. Although not a comprehensive analysis, it does provide an initial gauge of whether or not in-the-wild kineticons are successful and potentially improved upon.

The *Shake No* gesture was shown to be iconic of a "cannot complete" event and "needs attention" state. This dual meaning is appropriate given its current use in MacOSX (a password entered incorrectly). *Zoom* and *Suck In* were similarly successful as entrance and departure kineticons. *Bounce* was interpreted as a kineticon conveying "newly available", which is close to the MacOSX use. However *Rumble*, used in the iPhone to signal "movability" of icons, is less strong; it only scores well as a gross representation of "needing attention" and quite poorly in its intended "is movable" interpretation.

Correlations Between Interpretations

As noted several times in the results discussion thus far, it appears several of our interpretations were related conceptually. This manifested in the data as kineticons performing well in two categories. Indeed, our earlier suppositions are confirmed statistically (Pearson's correlations). As one would expect, "cannot complete" and "departure" are negatively correlated with loading ($\rho=-.21$ and $\rho=-.23$ respectively, $p<.001$). "Cannot complete" is also negatively correlated with the entrance interpretation ($\rho=-.24$, $p<.001$).

There were also several positive correlations. As hypothesized previously, "progress" appears to be conceptually similar to "loading/opening" ($\rho=.40$, $p<.001$). Also correlated were: replace/update and entrance interpretations ($\rho=31$, $p<.001$), and "cannot complete" and "no longer available" ($\rho=.36$, $p<.001$).

Applicability Across GUI Elements

There were two key behaviors we wanted our GUI element sets (application icon, menu item, dialog box, entire desktop) to exhibit if associated kinetic behaviors were to be considered successful. The first was that all four GUI element types performed well in a single interpretation (i.e., not ambiguous). Secondly, we wanted low variance in the strengths of that interpretation, which would suggest that a kineticon had similar expressive power across GUI types.

Encouragingly - as we believe this is one of the key benefits of kineticons - the results are strong. The four GUI elements with the *Heart Beat* kineticon clearly convey the "need for attention". This significantly out performs any other interpretations ($p<.05$). *Blowing in the Wind 2D* has an equally strong and singular result with the movable interpretation ($p<.05$). We designed *X-cross* to be a "cannot complete" behavior, an interpretation that was correlated with "no longer available". As such, the GUI element set with *X-cross* performs best in those two interpretations ($p<.05$ compared to all other interpretations, but not between "cannot complete" and "no longer available").

Conversely, the *Running* kineticon, which performed well as an application icon, yielding a strong "is movable" connotation, does not seems to generalize to other GUI elements. None of the positively rated interpretations are significant from each other, primarily due to high variance in participants' estimations of the interpretation strengths.

The high level result here is that kineticons can successfully generalize across a variety of GUI elements (3 of our 4 designs). However, this does depend on the specific design employed. Even kinetic behaviors that are extremely iconic when applied to some GUI elements, could have even a reverse connotations when applied to a different item. Thus, careful testing and iterative design is a necessity.

CONCLUSION

We have presented a definition of kineticons, or motion-based icons, and placed them into a taxonomy along side conventional graphical icons and those with animated graphics. Significantly, kineticons can be applied in concert with other iconographic schemes, including other forms of animation. Moreover, we show kineticons can be applied to and reused for a variety of elements in graphical user interfaces. Our evaluation yielded an initial set of effective kinetic behaviors and let us better gauge the effectiveness of kineticons that exist in popular platforms, like the iPhone.

ACKNOWLEDGMENTS

This work was supported in part by a Microsoft PhD Fellowship and grant IIS-0840766 from the National Science Foundation.

REFERENCES

1. Agarawala, A. and Balakrishnan, R. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proc. CHI '06*. 1283-1292.
2. Baecker, R., Small, I. (1990). Animation at the interface. In: B. Laurel (Ed.) *The Art of Human-Computer Interface Design*. Addison-Wesley, Reading, MA.
3. Baecker, R., Small, I., and Mander, R. Bringing icons to life. In *Proc. CHI '91*. 1-6.
4. Barclay, C.D., Cutting, J.W., and Kozlowski, L.T. Temporal and spatial factors in gait perception that influence gender recognition. *Perception and Psychophysics* 23, 1978, 145-152.
5. Bartram, L. Perceptual and interpretative properties of motion for information visualization. In *Proc. NPIV '97*. 3-7.
6. Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., and Ramos, G. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. In *Proc. UIST'06*.169-178.
7. Baumann, K. and Thomas, B. (2001). *User Interface Design of Electronic Appliances*, 1st Ed. CRC Press.
8. Beaudouin-Lafon, M. and Conversy, S. Auditory illusions for audio feedback. In *Proc CHI '96 Companion*. 299-300.
9. Blattner, M.M., Sumikawa, D.A., and Greenberg, R.M. Earcons and icons: their structure and common design principles. *Hum.-Comp. Interac*. 4, 1 (1989), 11-44.
10. Bodner, R.C. and MacKenzie, I.S. Using animated icons to present complex tasks. In Proc. Conf. Centre For Adv. Stud. Collab. Res. IBM Press. 1-11
11. Brewster, S. and Brown, L.M. Tactons: structured tactile messages for non-visual information display. In *Proc. Australasian User interface '04*. 15-23.
12. Brewster, S.A., Wright, P.C. and Edwards, A. An evaluation of earcons for use in auditory human-computer interfaces. In *Proc. CHI '93*. 222-227
13. Chang, B. and Ungar, D. Animation: from cartoons to the user interface. In *Proc. UIST '93*. 45-55.
14. Chatty, S. Defining the dynamic behaviour of animated interfaces. In *Proc. IFIP TC2/WG2.7 '92*. 95-111.
15. Conversy, S. Wind and wave auditory icons for monitoring continuous processes. In *Proc. CHI '98*. 351-352.
16. Easterby, R. The Perception of Symbols for Machine Displays. *Ergonomics* 13, 1 (1970), 149-158.
17. Enriquez, M.J. and MacLean, K.E. The Hapticon Editor: A Tool in Support of Haptic Communication Research. In *Proc. HAPTICS '03*. 356-362.
18. Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. In *Proc. CHI '09*. 1655-1664.
19. Gaver, W. Auditory Icons: Using Sound in Compute Interfaces. *Hum.-Comp. Interac*. 2, 2 (1986), 167-177.
20. Gaver, W. The Sonic Finder: An Interface that Uses Auditory Icons. *Hum.-Comp Interac,* 4, 1 (1989), 67-94.
21. Gonzalez, C. Does animation in user interfaces improve decision making? In *Proc. CHI '96*. 27-34.
22. Hankinson, J.C. and Edwards, A.D. Designing earcons with musical grammars. *SIGCAPH Comput. Phys. Handicap*. 65 (Sep. 1999), 16-20.
23. Harrison, C. and Hudson, S.E. Texture displays: a passive approach to tactile presentation. In *Proc. CHI '09*. 2261-2264.
24. Jacob, R.J., Girouard, A., Hirshfield, L.M., Horn, M.S., Shaer, O., Solovey, E.T., and Zigelbaum, J. Reality-based interaction: a framework for post-WIMP interfaces. *Proc. CHI '08*. 201-210.
25. Johansson, G. Visual Motion Perception. Scientific American, 232 (1975). 76-88.
26. Kendon, A. (1988). How Gestures Can Become Like Words. In *Cross-cultural Perspectives in Nonverbal Communication*. Toronto, C.J. Hogrefe, 131-141.
27. Kolers, P. A. (1972). *Aspects of Motion Perception*. Pergamon Press Inc., New York.
28. Kosugi, D. and Fujita, K. How do 8-month-old infants recognize causality in object motion and that in human action. *Japanese Psychol. Research*, 44 (2002), 66–72.
29. Lasseter, J. Principles of traditional animation applied to 3D computer animation. In *Proc. SIGGRAPH '87*. 35-44.
30. Lee, J.C., Forlizzi, J., and Hudson, S.E. The kinetic typography engine: an extensible system for animating expressive text. In *Proc. UIST '02*. 81-90.
31. Lee, S.H., and Blake, R. Visual form created solely from temporal structure. *Science*, 284 (1999). 1165-1168.
32. Liang, J., Huang, M.J. Highlighting in Information Visualization: A Survey. In *Proc. IV '10*. 79-85.
33. Lodding, K. Iconic Interfacing. *IEEE Computer Graphics and Applications*. 3, 2 (1983), 11-20.
34. McCloud, S. (1993). *Understanding Comics: The Invisible Art*. Kitchen Sink Press, Inc., Northampton, MA.
35. McCrickard, D., Zhao Q. and Stasko, J. Exploring Animation as a Presentation Technique for Dynamic Information Sources. In GVU Technical Report, GITGVU-99-47, 1999.
36. Mulder, A. (1996). Hand Gestures for HCI. School of Kinesiology, Simon Fraser University, Technical Report 96-1.
37. Myers, B.A. The importance of percent-done progress indicators for computer-human interfaces. *SIGCHI Bull*. 16, 4 (Apr. 1985), 11-17.
38. Neri, P. Moroone, M.C. and Burr, D.C. Seeing Biological Motion. *Nature*, 395 (1998), 894-896.
39. Pelphrey, K.A., Mitchell, T.V., McKeown, M.J., Goldstein, J., Allison, T., and McCarthy, G. Brain activity evoked by the perception of human walking: Controlling for meaningful coherent motion. *Jour. Neuroscience*, 23 (2003), 6819-6825.
40. Pylyshyn, Z., Burkell, J., Fisher, B., Sears, C., Schmidt, W. and Trick, L. Multiple parallel accessing visual attention. *Canadian Journal of Exper. Psychology*, 48, 2 (1994), 260-283.
41. Shanmugasundaram, M., Irani, P., Gutwin, C. Can smooth view transitions facilitate perceptual constancy in node-link diagrams? In *Proc. GI '07*. 71-78.
42. Stasko, J.T. (1993). Animation in user interfaces: principles and techniques. In *User Interface Software*. John Wiley & Sons, New York, NY. 81-101.
43. Thomas, B. H. and Calder, P. Supporting cartoon animation techniques in direct manipulation graphical user interfaces. *Inf. Softw. Technol*. 47, 5 (2005), 339-355.
44. Thomas, B.H. and Calder, P. Applying cartoon animation techniques to graphical user interfaces. ACM Trans. *Comput.-Hum. Interact.* 8, 3 (2001), 198-222.
45. Thomas, F. and Johnston, O. (1984). *Disney Animation: The Illusion of Life*. Abbeville Press, New York, NY.
46. Wilson, A.D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. Bringing physics to the surface. *Proc. UIST '08*. 67-76.
47. Wolfe, J.M., Kluender, K.R., Levi, D.M., Bartoshuk, L.M, Herz, R.S., Klatzky, R.L., and Lederman, S.J. (2006). *Sensation and Perception*. Sinauer Associates, Sunderland, MA.