Sketch-Editing Games: Human-Machine Communication, Game Theory and Applications

Andre Ribeiro
JST, Erato, Igarashi
Design Interface Project,
1-28-1-7F, Koishikawa
ribeiro@media.mit.edu

Takeo Igarashi
JST, Erato, Igarashi
Design Interface Project,
1-28-1-7F, Koishikawa
takeo@acm.org

ABSTRACT
We study uncertainty in graphical-based interaction (with special attention to sketches). We argue that a comprehensive model for the problem must include the interaction participants (and their current beliefs), their possible actions and their past sketches. It's yet unclear how to frame and solve the former problem, considering all the latter elements. We suggest framing the problem as a game and solving it with a game-theoretical solution, which leads to a framework for the design of new two-way, sketch-based user interfaces. In special, we use the framework to design a game that can progressively learn visual models of objects from user sketches, and use the models in real-world interactions. Instead of an abstract visual criterion, players in this game learn models to optimize interaction (the game's duration). This two-way sketching game addresses problems essential in emerging interfaces (such as learning and how to deal with interpretation errors). We review possible applications in robotic sketch-to-command, hand gesture recognition, media authoring and visual search, and evaluate two. Evaluations demonstrate how players improve performance with repeated play, and the influence of interaction aspects on learning.

Author Keywords
Sketch recognition; Machine Learning; Active Learning; Game Theory; Communication; Cooperation.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces: Theory and methods.

INTRODUCTION
When a machine learns to understand the symbol "ball!", it is learning an arbitrary convention, one that exists in the minds of a community of English speakers. Acting together, we establish with each other symbols that allow us to solve complex problems in an efficient, truly collaborative manner. Because individuals have no direct access to each others' thoughts, they must communicate to coordinate their distinct mental states and get them to converge (to some extent) in order to work together successfully. This process of "alignment" is necessary even when the parts are rational, cooperative, speak the same language, share much of the same knowledge and culture, but it is essential when they have asymmetric knowledge – such as in current Human-Computer Communication.

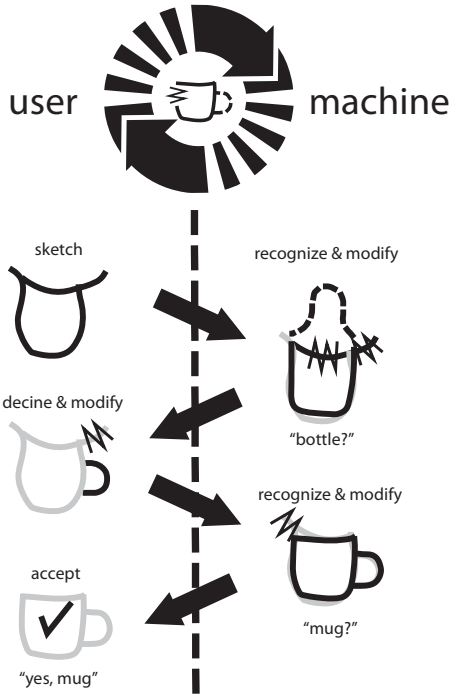


Figure 1. Turn taking in the Sketch-Editing Game.

In this article, we formulate a game theoretical model that captures some important aspects of this difficult problem. We introduce the model with past Human-Computer Communication examples and discuss its application in the learning and recognition of graphical symbols, where a user and a machine take turns modifying a joint sketch, *Fig.1*.

UIST '12, October 7–10, 2012, Cambridge, Massachusetts, USA.

The goal is to establish a two-way symbolic "dialog" between the user and the machine in which the machine not only routinely learns response to cues but also actively creates and manipulates symbols in interaction with humans.

This is an example of Interactive Learning. The problem has particular features not generally found in other learning problems. It consists of individuals that are learning about a process in which others are learning. And a learner in this situation must somehow consider not only the states of a physical process (*medium states*) but also the internal states of others (*knowledge states*). Additionally, learning must happen interactively (may involve turn-taking), transparently (players can observe and test what other's have learned) and be grounded in the common environment.

THE GAME

Suppose, for example, that a user has three types of objects in its kitchen - mugs, glasses and bottles - and that a robot can pass him the objects he sketches. This defines four subgames: need-a-mug, need-a-glass, need-a-bottle and need-nothing. This can be seen as a *coordination* game because user and machine are better off playing the same subgame. The problem is to jointly determine the intended subgame, without the assumption of any prior, common knowledge about these objects (in special, that the robot knows what "mug", "glass" or "bottle" are). A final solution is a symbol, a joint representation between them that unambiguously distinguishes the subgame for both. A solution is reached by a succession of symbolic proposals and modifications. A strategy for each player, at a time, is a specific modification of the current (joint) drawing that makes it less ambiguous (i.e., given the player's current beliefs). Modifications can be contour deletions (scratching), completions (drawing) or substitutions (morphing) – and they, together, are meant as a "is this what you mean?" query. We take this game to be a learning game with many rounds; where each player observes the opponents' moves and adapts its strategy until both are no longer uncertain about the reference. Symbolic communication is then a learning game where players learn better and better each other's references. We suggest that minimizing the expected number of interactions leads to both an attractive new medium for communication, and an effective training procedure.

Seen this way, learning and recognition are not artificially separated. The machine actively prompts the user for specific information whenever it recognizes a specific self-consistency gap (ambiguity) in its model. With better models, less and less interruptions are necessary (and learning transitions into use).

The game is played over a graph by two players, User (*U*) and Machine (*M*). The graph's nodes are shapes and edges are operations a player might perform on the source node (transforming it into the target). A path in the graph is then a progressive modification of some visual structure or structures. Games are often described by a tree (each subgame, a subtree). Imagine then that *M*'s prior knowledge consists of all previous games with *U*. In each round of the current game, *M* proposes a subgraph of this graph (i.e., a subgame) as a way of reducing uncertainty about what subgame is being played.

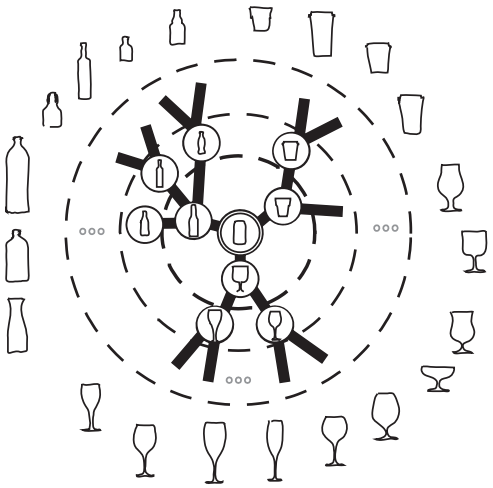


Figure 2. Game example.

Fig.2 depicts *M*'s graph after several games (sketchings of cans, bottles, and glasses shown in the graph's periphery). At the graph's center (double-circled) is *U*'s most recent sketch. The first set of transformations offered by *M* is seen in depth-1 nodes (first level). They are: adding a "neck" (towards bottle examples), adding a "base" (towards wine glasses) or widening the top (towards cup). The user can accept one of these suggestions and (optionally) use the transformed shape as basis for further editions. The next level shows subsequent transformations if each of these shapes were accepted.

As more is edited, more and more alternatives (or hypotheses) are ruled out as possible. Playing this game, *M* learns new symbols by learning objects' "boundaries" (how they can be deformed and still receive the same name) and coming to share those boundaries through interaction with other individuals. This way, communication serves to make interaction more efficient. A central suggestion in this article is that (visual) symbols are learned to optimize interaction. The first-level shapes in *Fig.2*, for example, make both players imagine an often used shape at a time. While something less likely (like "very tall cup") requires more words, or strokes, for players to jointly construct.

In the next section, we place the work in context. We then turn to the model (first motivating its central concepts and then fully formulating them). After that, we describe an

implementation of the model and its possible applications. We finally evaluate two of these applications, with an emphasis on learning performance.

RELATED WORK

The central problem relates to uncertainty and interaction, and thus Machine Learning (ML) and Human-Computer Interaction (HCI). We believe that the natural domain for HCI researchers interested in learning is not unsupervised learning (as sometimes taken as synonymous with Machine Learning), but semi-supervised or active learning [1,2]. Generally, unsupervised algorithms attempt to devise the most accurate classifier according to a provided exemplar set [3,4]. Active learning is of interest when the machine can, instead, query the user (or a database) for the classification of a given exemplar; and the intelligent choice of queries can improve the learning accuracy and/or speed. The user's response is often called a *labeling* of the exemplar. Consider the case of shape matching. A matching is often used as a shape classifier by starting with *U*'s shape (or sketch) and matching it against *M*'s (model) exemplars of different object classes. The resultant (distance) measures lead to a classification decision. In what follows, we consider the problem in the opposite direction: which exemplars should *M* suggest (or query) *U* at a time, to discover the correct class. The labeling (correct shape or not) of each suggestion informs, in this case, subsequent ones and allows *M* to change the shapes it is suggesting progressively, reactively, and quickly.

There has been a surge of research [9,7] tying machine learning techniques and vision problems in the past decade. Some researchers have, in turn, advocated the study, and discussed the power and difficulties of bringing these techniques to everyday computer use and HCI research [5,6]. For example, in [6] authors focused on the efficiency issue. They attempted to make a full re-training *unsupervised* cycle quick enough for interactive training for color and texture classification. For that, they used the popular "bag-of-feature" model [7,8,9] to represent objects (which ends up using over 1000 low-level image features to devise a classification decision). This rendered the interaction as a train-test cycle in a black-box fashion, *Fig.2c*. In this article, we explore an alternative connection between ML and HCI. Besides bringing an active learning perspective and game theoretical concepts to the problem, we address the issue of representation in learning, and we make the representations being learned observable and manipulateable by the user, *Fig.2d*. For the "bag-of-features" approach this seems plain unworkable. *Fig.2a* illustrates the first three features (learned with [8]) to detect cups such as the one in the top. The boxes are Haar-features, where the model expects paired high and low density of edge points. There are not only too many features, each usually at different scales, but few of them have any clear high-level interpretation.

Because ML is often framed in that language, the feature-set approach represents a straight-forward, out-of-the-box application of ML. To better address the unique constraints posed by HCI (semi-supervised, online, efficient, visual and intuitive representations), we believe we have to return to the concept of shape, which (due to the practical success of the feature-set approach) have lost some of its focus in vision. Although maybe not necessary for offline just-detect-the-object technology, it seems that whenever a human user is involved, so should the concept of shape. It's not yet clear, however, how to properly insert shapes into learning. The Shape-Editing game studied here uses the game formulation below and a new shape descriptor [28] to explore such issues in the intersection of shape representation and interactive learning. The descriptor attends to features relevant to HCI. It emphasizes shape regularity and the "gist" (instead of the feature) level of recognition. It represents shapes with a constructive process that recursively edits line-segments, *Fig.3e*. The process constructs rectangles, rectangles with one "bump" (or depression), two "bumps", etc. which are often related to the recognition of affordances [36].

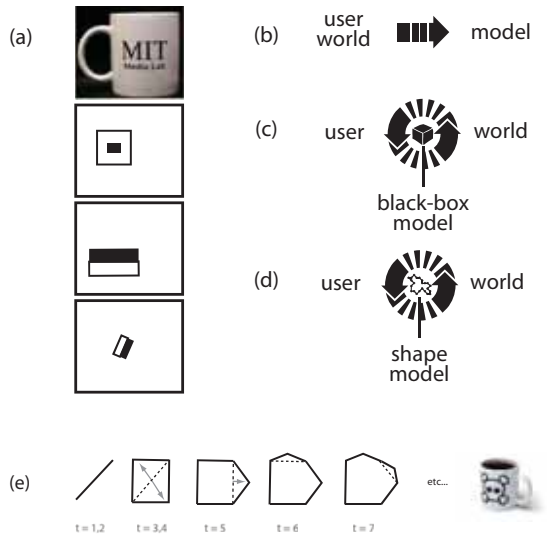


Figure 3. (a) first haar-features for a CART-tree, (b) batch data collection games, (c) iterative data collection and testing scheme, (d) shape learning scheme, (e) shape descriptor.

This top-down descriptor offers an attractive compromise between scalability and accuracy. One implementation [28] can match a shape to over 500 others under a second with accuracy within 5% of the state-of-the-art. We think specialized data-structures could scale the solution further, towards very large datasets (e.g., 1,000,000 shapes), offering a shape-matching alternative (and sketch-editing) to current index-based search solutions [10].

Following the unsupervised paradigm, games have been used as a setting to acquire ground-truth data for images

[11,12] and sketches [13]. After collected, data is then to be used as input for an unsupervised algorithm of choice. The process is one-way and non-interactive (*Fig. 2b*), and the model being constructed drives in no way the data being collected. We have observed that the process is dramatically accelerated when the model's uncertainty is clearly visualized by the user. This way, the interaction is an integral part of the solution (as opposed to the final-product's uninteresting last step). An exception comes from work in automatic translation or transcription [14], which often does have both the model and the user in the loop.

In the online formulation that follows the specific performance measure (otherwise traditionally the error rate [2]) is substituted by the game length (the expected number of interactions in the game). Optimization of large, *perfect*-information games have been studied in networking domains [24,25], while learning in games to a much lesser extent [26]. Models for games of incomplete information remain "laboratorial" (typically non-extensive), dating to Lewis' original signaling game formulation [16]. We studied large coordination games among low-income users, solving common problems, with cooperative (coalitional) game theoretical concepts with very positive results [27]. We are unaware of any other practical use of cooperative concepts, such as the Shapley value [23] used below. The relatively small quantity of game-theoretical concepts in HCI research is surprising, since it studies, by definition, a multi-party (cooperative) phenomenon.

Lee et al. [29] frame human-robot interaction as a "game" but say little about learning, uses no game theory, starts with idealized features, and is mostly descriptive. The solution is similar to work with no "games". Our basic premise (next section) is that people cope with joint problems by adopting hypotheses of what others are thinking and revising these hypotheses with interaction. This adds "cognitive" and "closed-loop" dimensions to general games, and leads to an explicit optimization criterion that doesn't require external models (e.g., markovian matrices) and that is related to principled game theoretical concepts.

In a broader scope, we place importance in mixed-initiative interaction [15], the study of interfaces that support efficient, natural interleaving of contributions by people and computers, aimed at converging on solutions to problems. Our main interest in this direction is to explore a new framework (targeting the visual, instead of the usual conversational medium). The framework developed allows machines to be positioned in a more active stance, ready to obtain information that they need (in as few queries as possible), as opposed to simply receiving information.

The game model proposed takes the user as an adapting creature, and we are interested in user learning (as opposed to assuming that only the machine is changing with interaction). A central motivation in the Shape-Editing game is to study more approachable ways for users to analyze, visualize and modify learned models. We see our work as an extension of efforts in that direction, in special in gesture-based interfaces [17,18]. James and Novins [19] study morphing animations in recognition, aimed at visualization, Lee et al. [20] overlay a cloud of ("completion") strokes recovered from large image datasets as guides for freehand drawing, and Igarashi et al. [21] suggest online candidate interpretations as result of fixed beautification procedures (not editing) for geometric design. The interaction is single-pass and fixed, and does not involve the back-and-forth construction of mutual understanding around complex models. The work here generalizes and enhances these paradigms.

Finally, we explore the issue of representational grounding, how can we represent the user's knowledge, without pre-encoding it in some way? The key seems to be to evolve that knowledge progressively and in collaboration, such that what the user refer to is always grounded in the environment – and so *U* and *M*'s knowledge can be progressively aligned (i.e., made increasingly similar) through actions in the real world. Work in this direction comes especially from Human-Robot systems. For example, the goal of the Ripley system [30] is to support collaborative human-robot interaction; however, most of the work so far has concentrated on the development of fixed mental models (e.g., perspective taking), involving very little learning and no across-interaction optimization.

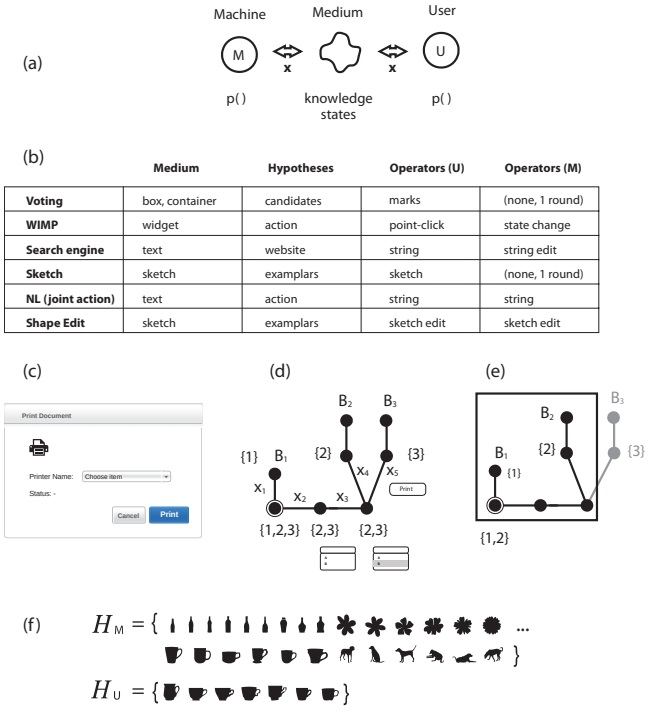
HUMAN-COMPUTER COMMUNICATION EXAMPLES

In this section we illustrate how typical Human-Computer communication (HCC) problems can be seen as games. We use the examples listed in *Fig.4b* to motivate and introduce key concepts. We do not study the game here as a general model for HCC, but feel that considering past HCC approaches is essential to better present and contextualize the work.

We see that cooperation, whatever its setting, involves a progressive modification of some structure (the medium). The main components of the model (*Fig.4a*) are two players (and their *knowledge states*) and a medium (and the *medium state*). The crux of the problem for players is that they can play many different games, and must coordinate unequivocally which they want to play at a time, through the medium. We call each a *subgame* to differ from the overall game. A player initially has incomplete information about which subgames the other wants to play. To reduce uncertainty, he changes the medium in some way so as to signal his intentions. The other respond similarly, and turn-taking takes place until they are ready to play a common subgame. Subgames are deterministic games with a fixed payoff structure, where players have shared and settled expectations and behaviors (no uncertainty).

This can be viewed as a series of subgame proposals made by the machine, followed by corrections made by the user, until an unambiguous solution is found for both. We

consider strategies for the players, each considering the other's *knowledge state*. The players' *knowledge states* are always subgames subsets (i.e., the set of subgames that a player is considering to play). Careful observation of successive structural modifications generally reveals a basic set of structural medium *operators* or editions (*x* in *Fig.4a*). The game's risk is the risk of miscoordination (or mis-reference) between players. We sometimes refer to a player's "opponent" to mean "the other player". And we call a user's edition (a correction of the current medium) an *error* in the game.



	Medium	Hypotheses	Operators (U)	Operators (M)
Voting	box, container	candidates	marks	(none, 1 round)
WIMP	widget	action	point-click	state change
Search engine	text	website	string	string edit
Sketch	sketch	examplars	sketch	(none, 1 round)
NL (joint action)	text	action	string	string
Shape Edit	sketch	examplars	sketch edit	sketch edit

Figure 4. (a) main elements, (b) example games, (c-e) WIMP example, (f) sketch example hypotheses-sets.

Ballot design is a very simple example of such a game. The medium is typically a paper formulary; the players are the voter and the official interpreting the ballot. The hypotheses-sets for both players are candidate sets (e.g., *{Bush, Kerry}*). The subgames are *vote-for-kerry* or *vote-for-bush*. The medium is typically graphical boxes or containers, and operators are marks. Poor designs lead to confusion and potentially miscoordination between a large numbers of voters and their government. Like in all examples here, the risk is a measure of visual confusion (that tells how well medium states visually distinguish the two hypotheses).

Voting is a simple example because it is a one round/turn game (uncertainty is, supposedly, resolved with a single action). Like voting, WIMP interfaces have also a very simple medium (designed for easy discrimination) and operators, but can have typically more rounds. Media are graphical widgets, and operators are mouse clicks (for *U*) and pre-coded state changes (for *M*). In the simple dialog box in *Fig.4c*, players can play the subgames *close*, *print-on-A* and *print-on-B*. The game's operators are used to construct a graph representation in the next section, *Fig.4d*. To play *print-on-A*, the user may start by clicking on the combo-box (use operation x_2), the machine respond by changing the widget to a popped state, the user click on "*A*" (x_3), the machine change the combo to unppoped (and possibly the "print" button to enabled), which the user clicks (x_4). To play this game *U* must also keep track of the machine's knowledge state (i.e., interaction deteriorate when states are not clear).

These are all graphically mediated interfaces. Search engine interfaces (e.g., google) are contrasting cases with a textual medium (a search string). Operators are often string-edit operations (character insertion, deletion and substitution). The case is interesting because the number of subgames is much larger, and the engine must effectively guess what's on the user's mind. The engine is adaptative (in the long term, across games) but there are typically few turns (within a game), with the engine simply listing its best, current mining matches.

The number of possible medium states is small in this case compared to graphical media. Gestural and sketch recognition interfaces deal with natural input and thus larger medium variation. Subgames are classes over gesture or sketch exemplars. The hypotheses for *M* in this case are the entire set of objects it "knows". The hypotheses-set for the user is the object he wants (or intends to sketch). This can be a set of exemplars of an object (e.g., a "cup") or a more specific set (e.g., a certain cup or any large cup). *Fig.4f* shows example hypotheses-sets for the former case.

Players specify what they are willing to play with their hypotheses-sets. In games here, *M* starts with a full set *H* and rule hypotheses out until it finds *U*'s playable subset in *H*. For the voting and WIMP games, *U* wants to typically play a singleton set (i.e., choose respectively one candidate or action). For others, *U* can specify (with its hypotheses set) a satisficing criteria (i.e., a set of acceptable subgames). The players' hypotheses-sets need not be identical, and learning is necessary if they are not. As result, we can say players "align" their representational sets through learning.

Sketch recognition is typically addressed by pre-training a gesture or sketch classifier, possibly from the two sets of exemplars. This is not convenient, however, when sets (e.g., what the user wants) change with the situation. Approaching the problem this way (offline, or in batch), learning also occurs outside of the interaction.

The previous examples differ in this way from Human-Human interaction where learning can emerge on-line, spontaneously, in response to interaction errors. Natural Language (NL) dialogue is an interesting example of a

game that, because of a very large number of subgames, relies on mechanisms of self-adjustment, turn-taking and repair [31]. With larger and more diverse graphical lexicons, errors are always possible and we can expect mechanisms that allow machines to similarly resolve graphical uncertainties online.

That requires both an online learning model and interaction design (i.e., how to present, accept and reject hypotheses online). We present a model next and then an interaction design for the case of sketches (designs for GUIs and other domains are not currently studied).

MODEL

The game is described by a Graph $G = \langle V, X \rangle$. Graph nodes V are medium states, edges X are medium operators. An edge $x \in X$ characterizes a (significant) difference between its two connecting nodes. Training (i.e., learning from previous games) is described by previous graph instances $G_1, G_2, \ldots, G_t$ and constitute the player's hypotheses-set, $H = \{G_1, G_2, \ldots, G_t\}$. The game is more naturally described in extensive form (as opposed to a single payoff matrix). The players' payoff is the game duration. Since players will stop playing only after they have resolved all their individual uncertainties, players must consider each other's knowledge states to minimize the game's duration. And they coordinate on paths with small expected duration for both.

Winning Paths

We first review the game's combinatorial structure, which is similar to [27]. We then introduce new concepts (next section) to formulate an optimal many-rounds game. Let h be the set of operators applied previously by a player (a "path" in G). A player's knowledge state $p(h)$ is a set of possible hypotheses or exemplars (e.g., different objects M thinks U might be thinking of, at a time). The knowledge state can be fundamentally described by a set function, $p : h \to 2^H$, denoting the set of possible hypotheses at a time (note this is merely a convenient functional form for a subset of H). This way, we endow a player with some private information, which is summarized by $p(h)$. Before applying an operator, the player then believes that what is truly being sketched is in the set $p(h)$. After the operation, the player reviews this set by observing the other's response, and so on.

Our overall approach is to first assume that the function $p(h)$ separates hypotheses perfectly (i.e., that no hypothesis is ruled-out erroneously). We then formulate and minimize a risk measure (which indicates the probability of misclassification). Humans often make decisions with nothing close to a complete statistical model. The emphasis here on model selection alleviates the often unpractical data-requirements of ML applications (e.g., [8] or [7] require hundreds of thousands of images, and sometimes days, to generate a classifier).

Let $p^U(h)$ be U's knowledge state after path h. A path is *winning* to M only if U has no more possible plays. That is, the game ends when the opponent has no further possible operations to apply to the medium, which happens when $p^U(h) = \{\}$, or schematically $h : p^U \to \{\}$. We then say two operations x_i and x_j are dependent for M if $p^U(\{x_i, x_j\}) = \{\}$. The set $W_t(H)$ of all winning paths of size t (in a game with hypotheses H) are the set of bases of a matroid [32,27] (which is a generalization of independence in vector spaces). Bases are the matroid maximal independent sets (i.e., sets that become dependent on adding any new element). The set of bases of all sizes up to T (a game parameter) is denoted simply $W(H)$.

The winning paths are an equilibrium solution in the game (i.e., neither player has incentive to deviate). Since the bases are the possible (final) plays in a game with hypotheses H, players must coordinate on one such path. They can estimate the opponent's likelihood of playing each path as a discrete probability distribution over the set of bases, $\sum_{B \in W(H)} P(B) = 1$.

The structure will allow to calculate efficiently several useful measures. Before devising the game's expected duration (payoff), reconsider *Fig.4d*. Name the subgames 1, 2 and 3 (*close*, *print-A* and *print-B*). M starts with hypotheses $\{1,2,3\}$, if U applies x_2, M reviews the set to $\{2,3\}$. Application of x_4 makes the set $\{2\}$. The path $\{x_2, x_3, x_4\}$ is a basis because the use of any other operation makes M's hypotheses set void. Other bases are $\{x_1\}$ and $\{x_2, x_3, x_5\}$.

Expected Length and Risk

With the bases-set $W(H)$, it's simple to calculate the expected duration, or length, L_1 of a game with hypotheses set H:

$$L_1(H) = \sum_{B \in W(H)} P(B) \cdot Length(B) \qquad (1)$$

where $Length(B) = |B|$ in this case[1]. The game's expected duration is given by its bases, which correspond to a set of paths in a graph. The expected duration of the game in *Fig.4d*, for example, is 1.67 (with $H = \{1,2,3\}$ and equiprobable bases).

This is the payoff of a game with fixed initial hypotheses-set, H. The expected duration can be drastically reduced in games where errors are possible (and informative). This is especially true in large games. In this case, players start with a subset of hypotheses, and, in case of errors, review the set (letting interaction feedback control the game).

[1] We assume that operators are associative and contiguous edges belonging to the same set of bases can be, at no cost, combined into a single edge (with length one).

Consider first a game with 2 rounds (and that "\" is typical set contraction). An error is the rejection (by the opponent) of a proposed operation *x*. The error can be informative, leading the player to reject the set of hypotheses $H \setminus p(x)$. An operation *x* can then be seen by *M* as a statistical test performed on *U* (with a risk $r(x)$). The min-expected length L_2 of a game with 2 rounds is then:

$$L_2(H) = 1 + \min_x [1 - P_e(x) L_1(p(x)) + P_e(x) L_1(H \setminus p(x))] \quad (2)$$

where $P_e(x)$ is the probability of *x* being rejected. The application of operation *x* splits the game in two, with probabilities $P_e(x)$ and $1 - P_e(x)$ of being played and lengths $L_1(H \setminus p(x))$ and $L_1(p(x))$. For example, in the game of *Fig.4d* consider that $P(B_2) \& P(B_3)$. In this case, *Eq.2* selects x_5. Players first assume that hypotheses are $\{1,2\}$ (*close* and *print-to-A*), and in case of error $\{3\}$. This corresponds to a design where the combo-box in *Fig.4c* has a default value (and *U* is forced to further operate the medium to reach the subgame *print-to-B*). Players agree (*Fig.4e*) then to carry the game over the subgraph with edges $\{x_1, f, x_4\}$ first, and move to the subgraph with $\{x_1, f, x_5\}$ as needed.

We can extend this solution by recursion to many rounds, $L_t(H)$, and to encompass the players' risk. In the second, the player, instead of incurring a cost of 1 at each round, incurs a risk $r(x)$. See [28] for a detailed discussion on the implementation of *Eq.2* and its elements in the visual domain (in special, shape operators and bases' risk and probability). This is a computational formulation of the game solution. We relate it now to a cooperative solution concept in game theory. Arrange *Eq.2* in the following way

$$L_2(H) = 1 + \min_x \{L_1(H) - P_e(x)[L_1(H) - L_1(H \setminus p(x))]\}$$
$$L_2(H) = 1 + \min_x \{L_1(H) - S(H,x)\}$$

where $S(H,x) = P_e(x)[L_1(H) - L_1(H \setminus p(x))]$.

Suppose the probability of error of an operation *x*, $P_e(x)$, is the sum of probabilities of bases that contain *x*. Then (see [34], proof of theorem 4.2):

$$S(H,x) = \sum_{B \in W(H)x} \frac{P(B)}{T - |B| + 1 \text{ if } \frac{T+1}{|B|}} [L_1(p(x)) - L_1(H \setminus p(x))]$$

In a game with 2 rounds, $S(H,x)$ corresponds to the game's Shapley value [23,35] for operation *x* over the defined matroid structure. Under this interpretation, the solution removes first the operation with lowest expected Shapley value. Players consequently assume a hypotheses-set with high Shapley value, considering the complementary set with lower value in case of error (requiring, in this case, more editions and thus minimizing the *expected* number of operations). The Shapley value is an ideal measure and generally too expensive to calculate. Due to the matroid structure, the value can be calculated efficiently [33,27].

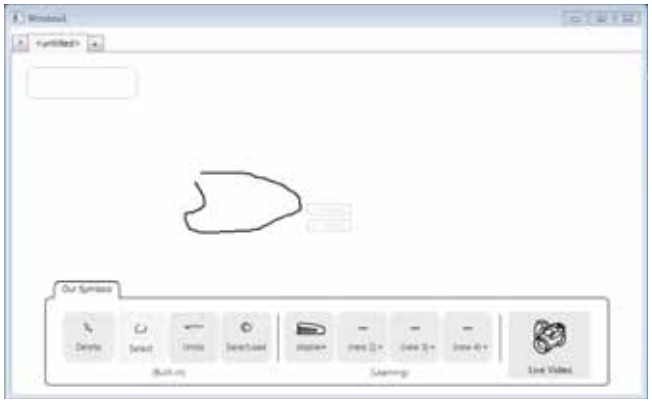


Figure 6. Prototype screenshot.

This view mirrors some intuitive aspects of human communication in general, where we assume at first the most likely aspects of a joint action and negotiate online, as needed, aspects that need alteration. The first-level shapes in *Fig.2* correspond to a reduced-hypotheses (or many-rounds) game. Reference to other (less likely) objects requires extra editions and further rounds.

PROTOTYPE INTERACTIVE SYSTEM

We have built a prototype interactive system to embody the game. *Fig. 6* shows a screenshot. It consists of a sketchboard and a model panel. While the game is played on the sketchboard, the lower bar visualizes the current models and shows recognition in live video.

As *U* sketches, *M* presents and updates in realtime its suggestions (shown as a pop-up button list to the side of *U*'s sketch, *Fig.6 light-gray*). Since each is an alternative set of shape operators, they are shown with morph-scratch-draw (substitute, erase, complete) animations on strokes. On hover, the user gets a quick morph (with fade) so he can quickly examine his choices. Each pop-up button corresponds to a set of hypotheses in the (reduced) game selected by *Eq.2* (and in the order of Shapley values). The user can erase or complete contours at any time, observing how changes affect *M*'s hypotheses-set and ordering. By choosing one interpretation, the user rejects all others. Optimization is carried online, stroke-by-stroke (as opposed to a sketch-and-submit model). The user's scratch gesture is pre-trained and cannot be changed.

Either *M* or *U* can declare the end of the game at any time. *M* declares the end when its hypotheses-set is empty and there are no more possible suggestions. *U* declares the end by choosing an additional ("new") hypothesis, which is always offered. In both cases, *M* offers *U* to add the current sketch as a new hypothesis and name it.

USE CASES

We now discuss use cases we have explored for the prototype.

Mobile Robot Sketch-to-Command

M is a mobile robot. *U* sketches a command for *M* to execute. Initially, *U* draws something that carries no meaning to the robot. Through learning, they can establish a common symbolic lexicon. We have prototyped the case where the user can sketch objects, locations, and arrows (*Fig. 7a*). The first two refer to live visual objects in front of both players, and the third to a single action (move source-to-destination). *Fig. 7b* shows data (hypotheses) collected in this scenario for household objects. *Fig. 7c* shows the 4 suggested transformations to the ambiguous user sketch in *Fig. 7a*. An interesting aspect of this scenario is that joint attention, like in human communication, can accelerate dramatically the reference game. Users can "snap" the diagrams to reality, or play the game "grounded" (i.e., *M* only disambiguates among recognized objects – or objects that it "sees"). Although we, for example, may know the shapes for a multitude of objects, there are few that we expect to see in particular joint action situations and places.

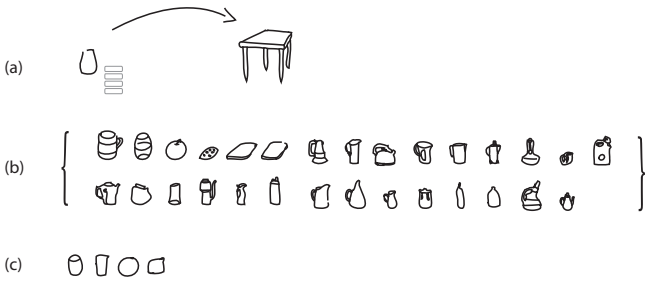


Figure 7. (a) sketch to command, (b) hypotheses, (c) suggestions.

Hand Gesture Recognition

The prototype can be readily used for pen or mouse gesture recognition. We have observed that allowing users to examine the variations or possible deformations in their gestures (similar to Bau's Octopocus technique [22]) makes the process more transparent. This is true for training and for use. The issue of recognition error is important and has been recognized for gestures [18]. The work here offers an alternative way to visualize the error and use it for learning.

Cooperative Sketch Completion

We studied games where users take turns completing sketches "forward". *M* learns such completions and use them in new games (i.e., with the next players). This way, with each new player, the shape editions grow richer and more imaginative and *M* constructs progressively, from playing, a "catalogue" of possible editions. As a first step, we designed a game where *M* starts with a number (*Fig.8*) and *U* is asked to creatively complete it. Children find playing this game after some training a lot of fun.

Figure 8. cooperative sketching.

Media Authoring

We have also used the prototype to assist users draw and animate. To help users draw, we foresee games where *U* draws something (e.g., a circle), and *M* suggests next steps (e.g., add a neck or a leg). *Fig.9a-b* shows editions learned from users drawing animals. To help users animate, we applied the previous framework to shape interpolation. In this case, animations were not only the system's means of communication, but also the final output. Each one of *U*'s deletions was taken to be a new model object. *Fig.9c* depicts an example: *U* drew the first frame, then deleted the arrow (a new model is created); he then drew only the arrow's shaft, which *M* completed with feathers. This was repeated a few times to make 1-stroke keyframes for the arrow wobbling animation. Since animation often compromises the recursive modification of sketch parts, learning parts' models can be useful.

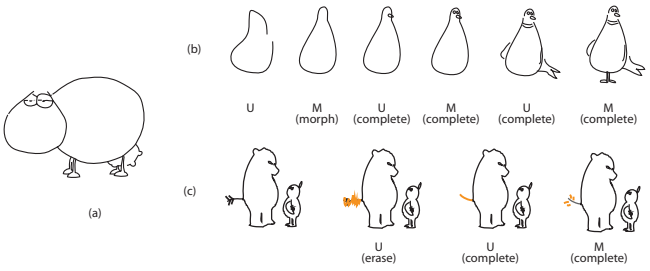


Figure 9. (a,b) assisted drawing, (c) animation.

Interactive Visual Web-Query

We have finally used the prototype to investigate a simplified version of the web search game (*Fig.4b*). Users sketch queries and ambiguity is resolved interactively (with sketch editing). For illustration, consider a simple example. We googled the terms "apple", "tomato" and "lemon" and selected as *M*'s possible hypotheses the top 3 related queries in each case. We used the goggle hit counts as probabilistic priors (see [28]). *Fig.10a* shows one run and *Fig.10b* lists *M*'s hypotheses with subsequent rounds. As in the previous games, the editions are learned from previous

users' sketches. In the next section, we evaluate this and the sketch-to-command scenarios in further detail.

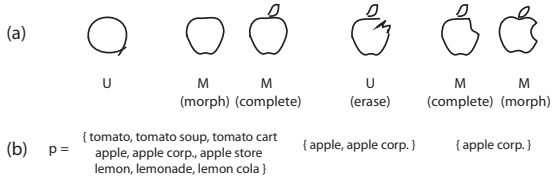


Figure 10. (a) query example, (b) *M*'s knowledge states .

EXPERIMENTS

We study learning and sketch-editing in two user trial studies with 70 participants playing 2 games. Our main goal is to subject the model to actual use – in special, to elucidate how many turns would prove necessary (in avg.) after some initial training and with natural input (which speaks directly to the model's applicability). Other relevant side discussions are also offered: that the designed interaction has an observable effect on these results, a curious benchmark on how humans would play the game, and a very favorable qualitative user evaluation for the web search scenario.

Machine Player (M)

We ran pilot user studies for the sketch-to-command and visual web-query use cases. The goal of these first experiments was to assess *M*'s longitudinal performance across different interactions and games. In the first condition (*U-M*), users played instances of the two games (15 household objects and web query trials each), and we examined the game duration across time (or users). With learning, we expected a progression of reducing game durations. For a baseline, (paired) users played the same games (*U-U*).

Participants were volunteers, aged 17-28, with little or intermediary tablet experience. Before the trials, they were given a 20 minutes training session on its use and the scratch gesture. At the end they were asked to complete a set of 5 exercises completing and erasing a set of shapes. The experiment was performed on two quad-core 2.8 and 2.6 GHz PCs, each connected to a Wacom Cintiq 12WX pen display.

For *U-M*, users were told to imagine they were playing a game where a computer would try to guess an object (or web query) from his/her sketches. They were told to help as much as possible, but that they could end the game at any time. They were asked to press one button for that case and another for a correct guess. Players were told that writing was against the game's rules (including numbers). No further instructions were given, and the experimenter did not answer questions during sessions. Before beginning, participants received a printed list of objects or queries in the games (which they were asked to read beforehand). The list was made available to account for the experimental constraint that *M* only discriminates among objects in the experiment (while *U* in a *U-U* trial, without the list, would consider a much larger number).

U received *M* feedback (hypotheses' labels and editions) with each of his/her strokes, but limited in the trial to a single suggestion. The game ended when either *M* guessed the object correctly, or, *U* or *M* ended the game. In the latter case, the trial was declared *learning-only*.

The *U-U* game reproduced *U-M*'s setup with a Wizard-of-Oz design. The (randomly assigned) initiating user proceeded stroke-by-stroke. The non-initiating user was asked to guess (label) and/or make any number of corrections to the other's sketch (editions). In *U-M*, *M*'s feedback was presented in less than 2 seconds (after the end of a stroke), while *U* in *U-M* and *U-U* could take any time. Note that this experiment's goal was to study *M*'s performance and not *U*'s perception (of the game). Users could not see each other, only the *GUI* in *Fig.6*, with added "end" and "correct!" buttons, and a "your turn" indicator. Since the response time is very different, we use the number of gestures (and not the more usual time-to-completion) as performance measure. The measure also accounts for *U*'s gesture errors.

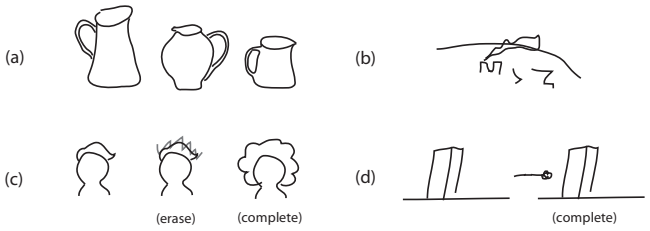


Figure 11. (a) object games examples, (b) learning-only trial, (c,d) web query edition disambiguation examples.

U-U had 20 participants (10 sessions), and *U-M* had 25 (25 sessions). A session consisted of the two games in succession (in random order). In the household objects game, 15 grasp/pushable objects in our lab with obvious names were selected. They were jar, camera, cell-phone, book, glasses, hat, knife, spoon, fork, stapler, plant vase, pen, crayon, tomato and orange. The order of the 15 objects was random. In the web query game, queries were selected (and randomly assigned) from the top 16 google searches in our location. They were "cerebral palsy", "Serena Williams", "World Trade Center", "Ronald Reagan", "college football rankings", "Jacksonville Jaguars", "9/11", "September 11", "flight 93", "mermaid", "Patriots", "Dr. Phil", "White House", "super bowl 2011", "presidencial debate" and "NHL". Two of the searches, "9/11" and "september 11", were merged into one.

Fig.12a shows the percentage of learning-only trials (i.e., without a correct guess) with time for the household object

game. *Fig.12c* shows the average number of gestures for the 15 objects over time. In this game, there is a trend towards 1-2 average gestures. The trend is present in *U-U* and emerges in *U-M* after a number of trials. More than one trial (i.e., exemplars) is often necessary to encompass different users' shapes and pose variations (e.g., *Fig.11a*).

Learning is "slower" in the web-query game (*U-M*), *Fig.12b*. It takes 9 users/sessions for *M* to guess correctly all queries for the first time (while only 3 for the household object game). The number of gestures, in both *U-M* and *U-U*, is (on average) also significantly larger for the web-query game, *Fig.12d*. Both observations can be explained by the abstraction level of concepts in this game (which leads to a larger variation on *U*'s depictions). *Fig.11c-d* highlights two trials, where users sketched to disambiguate "Ronald Reagan" and "Serena Williams" and "World trade center" and "11/9". *Fig.11b* shows a trial ended by *U* (learning-only) for "11/9".

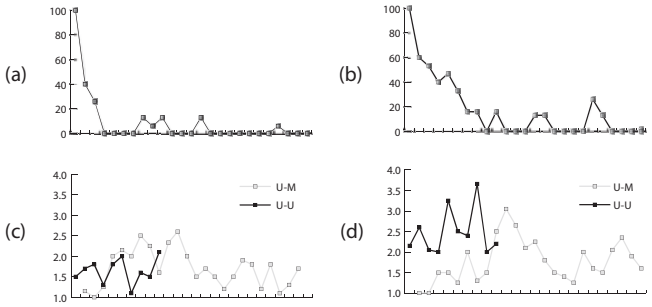


Figure 12. (a,b) average % of learning-only trials resp. in the object and web-query games , (c,d) average number of strokes.

The system is shape-bound and unable to use color and texture cues directly. One could argue that heavily articulated or amorphous objects (e.g., "coat") would take lots of exemplars to learn. Users seem to stick, however, to a surprisingly small set of (prototypical) shapes and poses, making the problem easier. This is arguably intrinsic to how we act together (and expected by the model).

User Player (U)

While the previous study addressed *M*'s learning performance, we addressed "user learning" and representational transparency next. We repeated the *U-M* setup in a two-factor study for 25 users. The control players played with word labels only (as opposed to labels and visual editions). Visual learning happens in the same way (but *M*'s editions are not visualized by *U*). *Fig.13* shows the results. A (paired) t-test for the percentage of correct guesses over trials indicates significant differences ($p = 0.0181$ and $p = 0.0104$) between the distributions in the two conditions, in the two games. The editions induce *U* to directly address *M*'s models (and often demonstrate relevant difference among objects or queries), which is useful to learning.

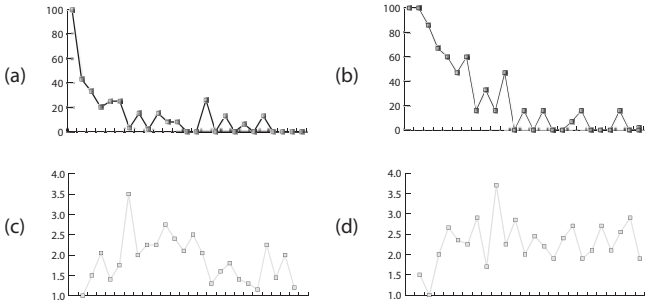


Figure 13. (a,b) average % of learning-only trials resp. in the object and web-query games , (c,d) average number of strokes.

These results point to an interaction between the representation and learning speed in the absolute. They favor the shape over the black-box model (*Fig.3c-d*) in the current tasks. That is, suppression of information about *M*'s current model (summarized by *M*'s editions) has immediate effect on learning performance. As implication, black-box models (where *U* cannot visualize *M*'s model in any practical way) miss an important aspect of the problem. These results suggest, in turn, the relevance of interaction aspects to the problem of learning. In the future, we plan to study learning efficiency across alternative representations.

The shape model has a direct effect not only on the resulting training data, but also on user engagement. A post-study survey asked participants to rank in a 5-point Likert scale if they would use the game to make a web-search (with 0 "not at all" and 5 "definitely"). Participants ranked the shape-model (Mean 4.69, SD 0.83) considerably more favorably than the black-box model (Mean 3.1, SD 0.95).

CONCLUSION

Everyday interaction is misleading in its simplicity. In our initial example game, unlike *U*, *M* does not know what "cup" means (a largely visual concept). Even when it does, does *U* mean that tall cup over there or the one with flowers? Uncertainty is part of any joint action, under any modality. We made the case for a way to address uncertainty in interaction (visually, in special). To articulate uncertainty visually ("tall? with flowers?") is often very powerful (we had users screaming/laughing in surprise, in web and drawing games) and natural (underlies all our GUIs). We demonstrated how such new sketch-based interaction paradigm can improve sketch-based learning. We will study further interaction and representation aspects of *U* and *M* learning performances. We believe that the work can also help understanding other modalities, machine mind-reading (and "writing"), and communication in general.

REFERENCES

1. Tong, S., Koller, D. Active Learning for Parameter estimation in Bayesian Networks. *NIPS* (2000).
2. Chaloner, K., Verdinelli, I. Bayesian Experimental Design: A Review. *Stat. Science* (1995).
3. Alvarado, C., Davis, R. Dynamically Constructed Bayes net for Multi-domain Sketch Recognition. *IJCAI*, (2005).
4. Ulgen F., Flavell C., Akamatsu N., Geometric Shape Recognition with Fuzzy Filtered Input to a Backpropagation Neural Network, *IEICE Transactions on Information and Systems*, 78, 2 (1995).
5. Maynes-Aminzade, D., Winograd, T., Igarashi, T. Eyepatch: Prototyping Camera-based Interaction through Examples, *UIST* (2007).
6. Fails, J., Olsen, D. Interactive Machine Learning. *ICUI (2003)*.
7. Opelt, A., Pinz, A., Fussenegger, M., Auer, P. Generic Object Recognition with Boosting, *PAMI* (2006).
8. Viola P., Jones, M. Rapid Object Detection Using Boosted Cascade of Simple Features. *CVPR* (2001).
9. Zhang, J., Marszalek, M. Lazebnik, S., Schmid C. Local Features and Kernels for Classification of Texture and Object Categories: a Comprehensive Study. *IJCV*, 73, 2 (2007).
10. Cao, Y., Wang, C., Zhang, L., Zhang, L. Edgel Index for Large-Scale Sketch-based Image Search. *CVPR* (2011).
11. Russel B., Torralba A., Murphy K., Labelme: A Database and Web-based Tool for Image Annotation, *IJCV*, 77, 1-3 (2008).
12. Von Ahn L., Dabbish L., Labeling Images with a Computer Game, *CHI (2004)*.
13. Johnson G., Do E., Games for sketch data collection, *Eurographics (2009)*.
14. Vidal, E., Casacuberta, F., Rodrigues, J., Civera, J., Martnez, C. Computer assisted translation using speech recognition. *IEE Transaction on Audio, Speech and Language Processing*, *14*, 3 (2006).
15. Horvitz, E. Reflections on Challenges and Promises of Mixed-Initiative Interaction, *AAAI Magazine 28* (2007).
16. Lewis, D., Convention. A Philosophical Study, Harvard University Press, Harvard (1979).
17. Bragdon, A., Zeleznik, R., Williamson, B., LaViola .. GestureBar: Improving the Approachability of Gesture-based Interfaces. *CHI* (2009).
18. Mankoff, J. Providing Integrated Toolkit-Level Support for Ambiguity in Recognition-Based Interfaces. *CHI* (2010).
19. James, A., Novins, K., Fluid Sketches: Continuous Recognition and Morphing of Simple Hand-Drawn Shapes. *UIST* (2000).
20. Lee, Y., Zitnick, L. Cohen, M. ShadowDraw: Real-Time User Guidance for Freehand Drawing. *SIGGRAPH* (2011).
21. Igarashi, T., Matsuoka, S., Kawachiya, S., Tanaka, H. Interactive Beautification: A Technique for Rapid Geometric Design. *UIST* (1997).
22. Bau, O., Mackay, W. OctoPocus: a Dynamic Guide for Learning Gesture-based Command Sets. *UIST* (2008).
23. Owen, G. Game Theory 3rd edition. Academic Press (1993).
24. McKelvey, R., McLennan, A. Computation of Equilibria in Finite Games. *Handbook of Computational Economics, Volume I* (1996).
25. Roughgarden, T. Selfish Routing and the Price of Anarchy. MIT Press (2005).
26. Kerns, M. Graphical Games. Algorithmic Game Theory, Cambridge University Press (2007).
27. Ribeiro, A. A Model of Joint Learning in Poverty: Coordination and Recommendation Systems in Low-Income Communities. *ICMLA* (2011).
28. Ribeiro, A., Igarashi, T. Joint Shape Editing Games: An Implementation. MIT Technical Report (2012).
29. Lee, K., Hwang, J. Human-robot Interaction as a Cooperative Game. O. Castillo (ed.) et al., *Trends in intelligent systems and computer engineering* (IMECS 2007).
30. Roy, D., Hsiao, K., Mavridis, N. Mental Imagery for a Conversational Robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 22 (2004).
31. Clark, H. Using Language. Cambridge Univ. Press (2006).
32. Oxley, J. Matroid Theory Oxford University Press (1992) .
33. Nagamochi, N., Zeng, D., Kabutoya, N., Ibaraki, T. Complexity of the Min. Base Game on Matroids. *Mathematics of Operations Research*. (1997).
34. Bilbao, J., Driessen, T., Jimenez-Losada, A., Lebron, E. The Shapley value for games on matroids: The static model. *Math. Meth. Oper. Res.* (2001).
35. Roth, A., The Shapley Value as a von Neumann–Morgenstern Utility, *Econometrica* 45 (1977).
36. Gibson, J. The Ecological Approach to Visual Perception. Lawrence Erlbaum (1979).