COGNITIVE MODELING

COGNITIVE MODELING

COGNITIVE MODELING —
DIAGRAMATIC REASONING

Formalizing Artistic Techniques and Scientific Visualization for Painted Renditions of Complex Information Spaces

Christopher G. Healey
Department of Computer Science, North Carolina State University
1010 Main Campus Drive, Raleigh, NC, 27695-7534
healey@csc.ncsu.edu

Abstract

This paper describes a new method for *visualizing* complex information spaces as painted images. Scientific visualization converts data into pictures that allow viewers to “see” trends, relationships, and patterns. We introduce a formal definition of the correspondence between traditional visualization techniques and painterly styles from the Impressionist art movement. This correspondence allows us to apply perceptual guidelines from visualization to control the presentation of information in a computer-generated painting. The result is an image that is visually engaging, but that also allows viewers to rapidly and accurately explore and analyze the underlying data values. We conclude by applying our technique to a collection of environmental and weather readings, to demonstrate its viability in a practical, real-world visualization environment.

1 Introduction

This paper describes a formal method for constructing visual representations of complex information spaces that support rapid and accurate exploration and analysis. Our technique falls within the domain of *scientific visualization,* the conversion of collections of strings and numbers (or datasets, as they are often called) into images that allow viewers to *explore, analyze, validate,* and *discover* within their data. We focused on two important issues [Smith and Van Rosendale, 1998] during our investigations:

1. *Multidimensional displays:* Our technique must support the visualization of multiple overlapping data fields together in the same display. This is much more difficult than representing a single data field in isolation. Designing techniques to effectively represent multidimensional datasets is an open area of research in visualization.

2. *Aesthetic displays:* We also seek to create images that are visually engaging. We believe this will motivate viewers to study a visualization in more detail. It will draw viewers into an image, and can be used to emphasize areas of importance in a dataset.

We address these goals by: (1) applying results from human perception to create images that harness the strengths of our low-level visual system, and (2) using artistic techniques from the Impressionist movement to produce *painterly renditions* that are both beautiful and engaging. From an AI perspective, the contribution of this work is the identification of a close relationship between specific painterly techniques and the performance properties of human perception; our formalization lays the groundwork for the generation of scientific visualizations that are effective and aesthetically pleasing.

Our technique converts a collection of data values into a painterly image as follows. First, one or more computer generated “brush strokes” are attached to each data element in the collection. A brush stroke has style properties (*e.g.,* color, length, or direction) that we can vary to modify its visual appearance. Data values in the data element are used to select specific states for the different properties. The result is a stroke that represents its corresponding data element. Rendering all of the strokes for every data element produces a painterly image whose stroke properties visualize the underlying dataset.

The remainder of this paper describes in detail how each step in this process is managed and controlled. We begin by defining formalisms for: (1) a multidimensional dataset and its visualization, and (2) the brush strokes that make up a painterly image. We next present a set of perceptual rules on the use of color and texture in visualization that we extend via our formalisms to the painterly domain. These rules ensure the images we produce represent a dataset in a perceptually salient manner. Finally, we discuss how our techniques were used to visualize a real-world collection of environmental and weather readings for the continental United States. We conclude with a summary and a short description of future work.

2 Formalisms

We began our investigation by identifying methods for building painterly images that we can use to represent multidimensional datasets. A key insight is that many painterly styles correspond closely to perceptual features that are detected by the human visual system. In some sense this is not surprising. Artistic masters understood intuitively which properties of a painting would capture a viewer’s gaze, and their styles naturally focused on harnessing these features. Moreover, certain movements used scientific studies of the visual system to help them understand how viewers would perceive their work. The overlap of artistic styles and perception offers a important starting point: the body of knowledge on the use of perception during visualization will help us to predict how corresponding painterly styles might perform in the same environment.

In order to make use of this advantage, we define a relationship between traditional visualization techniques and painted images. This is done by constructing a correspondence between formal specifications of the two environments. The correspondence can then be used to extend our perceptual guidelines to a painterly domain.

2.1 Multidimensional Visualization

A simple formalization of a multidimensional visualization consists of two parts: a description of the dataset, and a definition of the mapping function used to convert it into an image. A multidimensional dataset $D = \{e_1, \ldots, e_n\}$ contains n samples points or data elements e_i. D represents two or more data attributes $A = \{A_1, \ldots, A_m\}$, $m > 1$; data elements encode values for each attribute, that is, $e_i = \{a_{i,1}, \ldots, a_{i,m}\}, a_{i,j} \in A_j$.

Visualization begins with the construction of a data-feature mapping $M(V, \phi)$ that converts the raw data into images that are presented to the viewer. $V = \{V_1, \ldots, V_m\}$ identifies a visual feature V_j to use to display data attribute A_j. $\phi_j : A_j \rightarrow V_j$ maps the domain of A_j to the range of displayable values in V_j. Based on these definitions, visualization is the selection of M and a viewer's interpretation of the images produced by M. An effective visualization chooses M to support the exploration and analysis tasks the viewer wants to perform.

3 Painterly Styles

Our investigation of painterly styles is directed by two separate criteria. First, we restrict our search to a particular movement in art known as Impressionism. Second, we attempt to pair each style with a corresponding visual feature that has proven to be effective in a perceptual visualization environment. There are no technical reasons for why Impressionism was chosen over any other movement. In fact, we expect the basic theories behind our technique will extend to other types of artistic presentation. For our initial work, however, we felt it was important to narrow our focus to a set of fundamental goals in the context of a single type of painting style.

The term "Impressionism" was attached to a small group of French artists (initially including Monet, Degas, Manet, Renoir, and Pissarro, and later Cézanne, Sisley, and Van Gogh, among others) who broke from the traditional schools of the time to approach painting from a new perspective. The Impressionist technique was based on a number of underlying principles (see also [Schapiro, 1997]):

1. *Object and environment interpenetrate.* Outlines of objects are softened or obscured (*e.g.,* Monet's water lilies); objects are bathed and interact with light; shadows are colored and movement is represented as unfinished outlines.
2. *Color acquires independence.* There is no constant hue for an object, atmospheric conditions and light moderate color across its surface; objects may be reduced to swatches of color.
3. *Show a small section of nature.* The artist is not placed in a privileged position relative to nature; the world is shown as a series of close-up details.
4. *Minimize perspective.* Perspective is shortened and distance reduced to turn 3D space into a 2D image.
5. *Solicit a viewer's optics.* Study the retinal system; divide tones as separate brush strokes to vitalize color rather than graying with overlapping strokes; harness simultaneous contrast; use models from color scientists like Chevreul [Chevreul, 1967] or Rood [Rood, 1879].

Although these general characteristics are perhaps less precise than we might prefer, we can still draw a number of important conclusions. Properties of hue, luminance, and lighting were explicitly controlled and even studied in a scientific fashion by some of the Impressionists. Rather than using an "object-based" representation, the artists appear to be more concerned with subdividing a painting based on the interactions of light with color and other surface properties. Additional painterly styles can be identified by studying the paintings themselves. These styles often varied dramatically between individual artists, acting to define their unique painting techniques. Examples include:

- *path:* the path or direction a brush stroke follows; Van Gogh made extensive use of curved paths to define boundaries and shape in his paintings; other artists favored simpler, straighter strokes,
- *length:* the length of individual strokes on the canvas, often used to differentiate between contextually different parts of a painting,
- *density:* the number of strokes laid down in a fixed area of canvas,
- *coarseness:* the coarseness of the brush used to apply a stroke; a coarser brush causes visible bristle lines and surface roughness, and
- *weight:* the amount of paint applied during each stroke; heavy strokes highlight brush coarseness and produce ridges of paint that cause underhanging shadows when lit from the proper direction.

In this context, a painting P can be seen as a collection of n brush strokes $P = \{b_1, \ldots, b_n\}$, with each stroke made up of p style properties S_j, that is, $b_i = \{s_{i,1}, \ldots, s_{i,j}\}$, $s_{i,j} \in S_j$.

Although it would be tedious (and perhaps uninformative) to characterize a real painting in this manner, these definitions provide an effective way to relate the visualization process to a painted image. First, we can match many of the painterly styles to visual features used during visualization. For example, color and lighting in Impressionism has a direct correspondence to the use of hue and luminance in perceptual visualization. Other styles (*e.g.,* path, density, and length) have similar partners in perception (*e.g.,* orientation, contrast, and size). Second, data elements e_i in a dataset are analogous to brush strokes b_i in a painting. Attribute values $a_{i,j}$ in element e_i could therefore be used to select specific $s_{i,j}$ for each style in b_i.

Consider a data-feature mapping $M(V, \phi)$ in this context. The visual features $V_j \in V$ can be converted to their corresponding painterly styles S_j. M now describes how to convert a data element e_i into painted brush stroke b_i whose visual appearance represents the attribute values $a_{i,j}$ embedded in e_i. The close correspondence $V_j \leftrightarrow S_j$ between perceptual features and many of the painterly styles we hope to apply is particularly advantageous. Since numerous controlled experiments on the use of perceptual features have already been conducted, we have a large body of evidence to use to predict how we expect painterly styles to react in a multidimensional visualization environment.

4 Perceptual Characteristics

Past research has studied methods for applying rules of perception during visualization [Bergman *et al.*, 1995; Healey and Enns, 1999; Rheingans and Tebbs, 1990]. The cognitive abilities of the low-level human visual system have been studied extensively in the area of human psychophysics. One interesting result is the identification of a limited set of visual features that are detected rapidly, accurately, and relatively effortlessly by a human viewer [Triesman, 1985; Wolfe, 1994]. These features are similar to the ones we display during multidimensional visualization (*e.g.,* hue, luminance, orientation, size, and motion). When combined properly, they can be used to perform exploratory analysis tasks like searching for data elements with a particular attribute value, identifying boundaries between groups of elements with similar values, tracking elements as they move in time and space, and estimating the number of elements with common values. The ability to harness the low-level visual system during visualization through the use of these features is especially attractive, since:

- analysis is rapid and accurate, often requiring no more than 200 milliseconds,
- task completion time is constant and independent of the number of elements in a display, and
- different visual features can interact with one another to mask information; psychophysical experiments allow us to identify and avoid these interference patterns.

A data-feature mapping that builds on a perceptual foundation can support high-level exploration and analysis of large amounts of data in a relatively short period of time. Our recent work focuses on the combined use of fundamental properties of color and texture to encode multiple attributes in a single display. We draw on three specific areas of research in perception and visualization to guide the construction of our brush strokes: color selection, texture selection, and feature hierarchies that can cause visual interference and masking.

4.1 Color Selection

Color is a common feature used in many visualization designs. Some techniques attempt to measure and control the color difference viewers perceive between pairs of colors. This allows:

- *perceptual balance:* a unit step anywhere along the color scale produces a perceptually uniform difference in color,
- *distinguishability:* within a discrete collection of colors, every color is equally distinguishable from all the others (*i.e.,* no color is "easier" or "harder" to identify), and
- *flexibility:* colors can be selected from any part of color space.

Standard color models like CIE LUV or CIE Lab use Euclidean distance to approximate perceived color difference. More complex techniques extend this basic idea. For example, Rheingans and Tebbs [Rheingans and Tebbs, 1990] plotted a path through a color model, a allowed a viewer to vary how colors are selected along the path. Ware constructed color scales that spiral up around the luminance axis [Ware, 1988]; such a scale maintains a uniform simultaneous contrast error along its length. Healey and Enns [Healey and Enns, 1999] showed that color distance, linear separation, and color category must all be controlled to select discrete collections of equally distinguishable colors.

Our color selection technique combines different aspects of each of these methods. A single loop spiraling up around the luminance axis is plotted in the region of CIE LUV space that contains our monitor's color gamut. The path is subdivided into r named color regions (*e.g.,* a blue region, a green region, and so on). n colors are then selected by choosing $\frac{n}{r}$ colors uniformly spaced along each of the r color regions. The result is a set of colors selected from a perceptually balanced color model, each with a roughly constant simultaneous contrast error, and chosen such that color distance and linear separation are constant within each named color region.

4.2 Texture Selection

Although texture is often viewed as a single visual feature, it can be decomposed into fundamental perceptual dimensions. Research in computer vision has used properties like regularity, directionality, and contrast to perform automatic texture segmentation and classification. Results from psychophysics have shown that many of these properties can also be detected by the low-level visual system.

One promising approach in visualization has been to use the perceptual dimensions of a texture pattern to represent multiple data attributes. Individual values in a data element control a corresponding texture dimension, producing a texture pattern that changes its visual appearance based on the underlying dataset. Grinstein et al. [Grinstein *et al.*, 1989] built "stick-man" icons to represent high dimensional data elements; the orientation of each limb encodes a value for one particular attribute. Ware and and Knight [Ware and Knight, 1995] displayed Gabor filters that modified their orientation, size, and contrast based on the values of three independent data attributes. Healey and Enns [Healey and Enns, 1999] constructed perceptual texture elements (or pexels) that varied in height, density, and regularity; their results showed that both height and density were perceptually salient, but regularity was not. More recent work [Weigle *et al.*, 2000] found that an orientation difference of $15°$ is sufficient to rapidly distinguish visual elements.

4.3 Feature Hierarchy

A third factor that must be considered is visual interference, that is, a situation where one visual feature masks another. Although the need to rank each brush stroke style's perceptual strength is not necessary in a painting, this information is critical for effective visualization design. The most important data attributes (as defined by the viewer) should be displayed using the most salient features. Secondary data should never be visualized in a way that masks the information a viewer is most interested in seeing.

Perceptual features are ordered in a hierarchy by the low-level visual system. Results reported in both the psychophysical and visualization literature have confirmed a luminance–hue–texture interference pattern. Variations in luminance can slow a viewer's ability to identify the presence of individual hues or the spatial patterns they form [Callaghan, 1990]. The interference is asymmetric: random variations in hue have no effect on a viewer's ability to see luminance patterns. A similar asymmetric hue on texture interference has also been shown to exist [Healey and Enns, 1999; Triesman, 1985]; random variations in hue interfere with the identification of texture patterns, but not vice-versa. These results suggest that luminance, then hue, then various texture properties should be used to display attributes in order of importance.

5 Painterly Visualization

Based on the perception guidelines discussed above, and on our formal correspondence between visualization techniques and painterly images, we decided to build a system that varied brush stroke color, size, spatial density, and orientation to encode up to four independent data attributes (in addition to the two spatial values used to position each stroke). The presence of feature hierarchies suggest color should be used to represent the most important attribute, followed by the texture properties. The results of [Healey and Enns, 1999] further refine this to applying color, size, density, and orientation in order of attribute importance.

The brush strokes in our current prototype are constructed using a simple texture mapping scheme. A real painted stroke was digitized and converted into a texture map. This texture map is attached to a polygon to produce a reasonable approximation of a generic brush stroke. The stroke's position, color, size, and orientation are controlled by modifying the texture map and transforming the polygon. Density is varied by changing the number of strokes rendered in a unit area of screen space. Fig. 1 shows an example of brush strokes with four different greyscales, sizes, densities, and orientations (greyscale is used only for the printed figures; on-screen images are displayed in full color).

5.1 Practical Applications

One of the application testbeds for our visualization techniques is a dataset of monthly environmental and weather conditions collected and recorded by the Intergovernmental Panel on Climate Change. This dataset contains mean

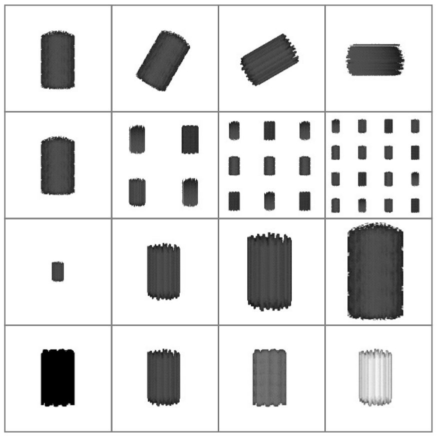

Figure 1: Examples of texture mapped brush strokes with different orientations (top row), densities (second row), sizes (third row), and greyscales (fourth row)

monthly surface climate readings in $\frac{1}{2}°$ latitude and longitude steps for the years 1961 to 1990 (*e.g.,* readings for January averaged over the years 1961-1990, readings for February averaged over the years 1961-1990, and so on). We chose to visualize temperature, precipitation, pressure, and windspeed. Based on this order of importance, we built a data-feature mapping M that assigns brush stroke greyscale (or color for on-screen images), size (or coverage), density, and orientation, respectively, to our four attributes. Temperature is represented by greyscales selected uniformly from a perceptually balanced luminance path. This path runs from dark (for cold temperatures) to bright (for hot temperatures). Precipitation is represented size (*i.e.,* the amount of an element's spatial region its brush stroke covers). Sizes range exponentially from very small coverage (for little or no precipitation) to full coverage (for heavy precipitation). Windspeed is represented by orientations ranging from $0°$ or upright (for weak winds) to $90°$ or flat (for strong winds). Finally, pressure is represented by four increasingly dense arrays of brush strokes: a single stroke, a 2×2 array of strokes, a 3×3 array, and a 4×4 array; continuous pressure values are discritized into four uniform ranges, then mapped to the appropriate density (sparse for low pressure, dense for high pressure).

Fig. 2 shows a visualization of data for February in the eastern half of the continental United States. Although unlikely to be mistaken for a real Impressionist painting, we feel the image contains important aesthetic qualities that make it stand out from a traditional visualization. The top four images (the top row of Fig. 2) use a perceptual greyscale ramp to show the individual variation in temperature, precipitation, pressure, and windspeed. M was used to construct the painterly visualization of all four attributes shown in the bottom image of Fig. 2. Various luminance and texture patterns representing different weather phenomena are noted in this image.

We have applied our painterly visualizations to a number of additional real-world environments including scientific simulations, e-commerce activity logs, and medical scans. Anecdotal feedback from domain experts collaborating on our ef-

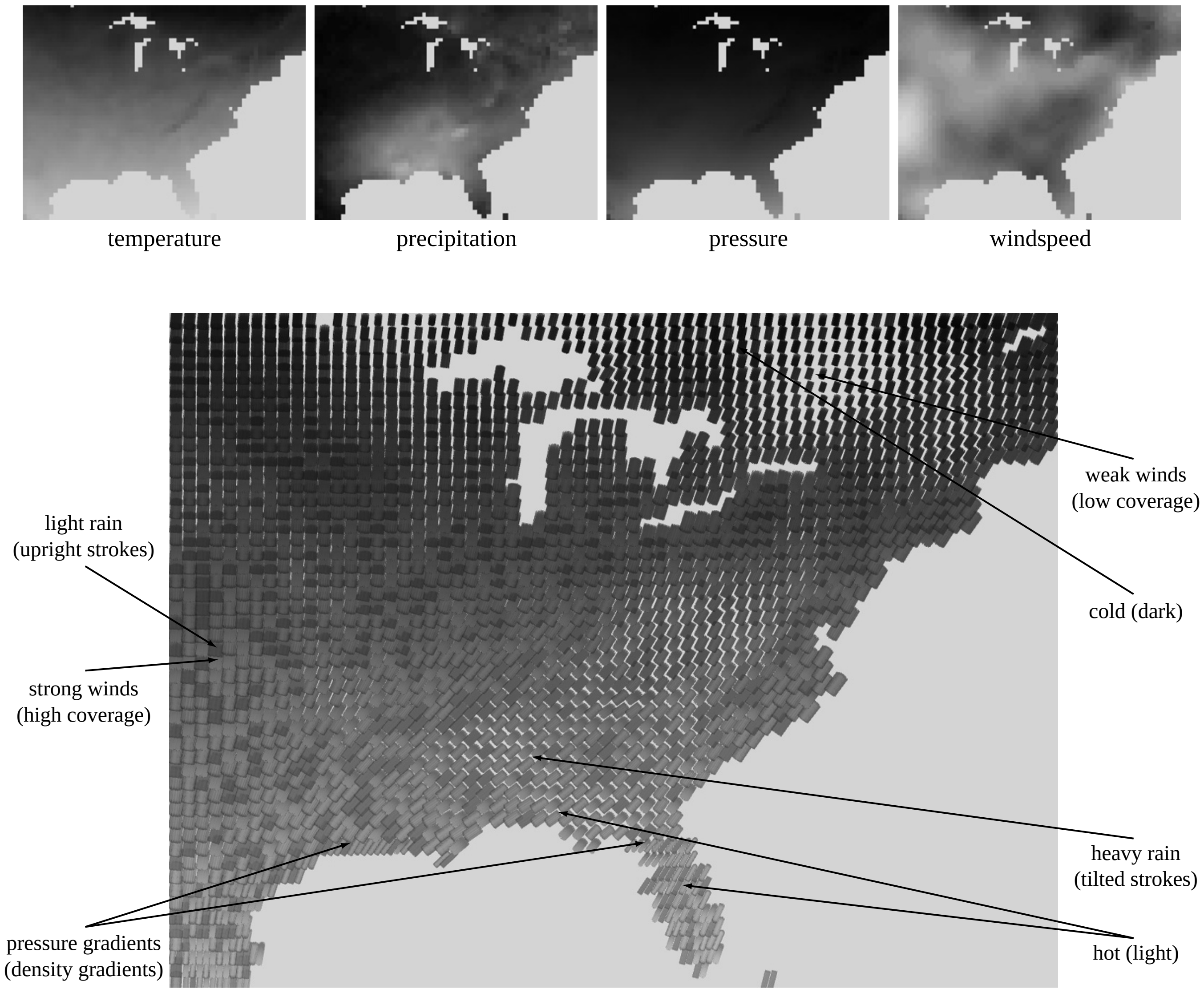
temperature
precipitation
pressure
windspeed
weak winds (low coverage)
light rain (upright strokes)
cold (dark)
strong winds (high coverage)
heavy rain (tilted strokes)
pressure gradients (density gradients)
hot (light)

Figure 2: A painterly visualization of environment conditions for February over the eastern United States: (top row) greyscale ramps (dark for small values to light for large values) of temperature in isolation, precipitation in isolation, pressure in isolation, and windspeed in isolation; (bottom image) a painterly visualization of all four attributes represented with the brush stroke properties greyscale, size (or coverage), density, and orientation

forts suggests that our technique achieves its goal of producing images that: (1) represent multidimensional datasets in a clear and effective manner, and (2) contain many of the aesthetic and visually engaging properties of a real painting.

6 Conclusions and Future Work

This paper describes a new method of visualization that uses painted brush strokes to represent multidimensional data elements. Our strokes support the variation of visual properties based in large part on styles from the Impressionist school of painting. Each attribute in a dataset is mapped to a specific painterly style; a data element's attribute values can then be used to vary the visual appearance of its brush stroke. The styles we chose are closely related to perceptual features detected by the low-level human visual system. Research studying the use of these features during visualization allows us to optimize the selection and application of the corresponding painterly styles. The result is a "painted image" whose color and texture patterns are used to explore, analyze, verify, and discover information stored in a multidimensional dataset.

One important area of future work is the construction of new brush stroke models. Texture maps are common in most graphics APIs, and are often rendered using hardware acceleration. Unfortunately, certain styles (*e.g.,* stroke coarseness or weight) are not easy to manipulate using texture maps. It may also be difficult to animate textured brush strokes during real-time visualization. We are currently investigating three potential solutions to this problem: (1) building a library of texture maps that explicitly differ across certain styles; (2) using mathematical spline surfaces to model more sophisticated brush stroke properties, and (3) using a physical simulation system to construct realistic strokes. Early results suggest a combination of models (*e.g.,* a texture map library whose entries are precomputed or dynamically updated) may be most appropriate.

We are also working to identify new painterly styles, and to integrate them into our stroke models. Two promising candidates we have already discussed are coarseness and weight. We are reviewing literature on technical and artistic properties in Impressionism, while at the same time searching for perceptual features that may correspond to new painterly styles. Increasing the number of styles we can encode in each brush stroke will allow us to represent larger datasets with higher dimensionality.

We note one final advantage we can derive from the correspondence between perceptual features and painterly styles. We measure the perceptual salience of a visual feature using controlled psychophysical experiments. Exactly the same technique can be used to investigate the strengths and limitations of new painterly styles. Just as research in perception helps us to identify and control brush stroke properties during painterly visualization, work on new styles may offer insight into how the low-level visual system "sees" certain combinations of visual properties.

Acknowledgments

This work is supported by the National Science Foundation research grant NSF-ACI-0083421.

References

[Bergman *et al.*, 1995] Lawrence D. Bergman, Bernice E. Rogowitz, and Lloyd A. Treinish. A rule-based tool for assisting colormap selection. In *Proceedings Visualization '95*, pages 118–125, Atlanta, Georgia, 1995.

[Callaghan, 1990] T. C. Callaghan. Interference and dominance in texture segregation. In D. Brogan, editor, *Visual Search*, pages 81–87. Taylor & Francis, New York, New York, 1990.

[Chevreul, 1967] Michel Eugène Chevreul. *The Principles of Harmony and Contrast of Colors and Their Applications to the Arts*. Reinhold Publishing Corporation, New York, New York, 1967.

[Grinstein *et al.*, 1989] G. Grinstein, R. Pickett, and M. Williams. EXVIS: An exploratory data visualization environment. In *Proceedings Graphics Interface '89*, pages 254–261, London, Canada, 1989.

[Healey and Enns, 1999] Christopher G. Healey and James T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[Rheingans and Tebbs, 1990] Penny Rheingans and Brice Tebbs. A tool for dynamic explorations of color mappings. *Computer Graphics*, 24(2):145–146, 1990.

[Rood, 1879] Ogden Nicholas Rood. *Modern Chromatics, with Applications to Art and Industry*. Appleton, New York, New York, 1879.

[Schapiro, 1997] Meyer Schapiro. *Impressionism: Reflections and Perceptions*. George Brazillier, Inc., New York, New York, 1997.

[Smith and Van Rosendale, 1998] P. H. Smith and J. Van Rosendale. Data and visualization corridors report on the 1998 CVD workshop series (sponsored by DOE and NSF). Technical Report CACR-164, Center for Advanced Computing Research, California Institute of Technology, 1998.

[Triesman, 1985] A. Triesman. Preattentive processing in vision. *Computer Vision, Graphics and Image Processing*, 31:156–177, 1985.

[Ware and Knight, 1995] Colin Ware and William Knight. Using visual texture for information display. *ACM Transactions on Graphics*, 14(1):3–20, 1995.

[Ware, 1988] C. Ware. Color sequences for univariate maps: Theory, experiments, and principles. *IEEE Computer Graphics & Applications*, 8(5):41–49, 1988.

[Weigle *et al.*, 2000] Christopher Weigle, William G. Emigh, Geniva Liu, Russell M. Taylor, James T. Enns, and Christopher G. Healey. Oriented texture slivers: A technique for local value estimation of multiple scalar fields. In *Proceedings Graphics Interface 2000*, pages 163–170, Montréal, Canada, 2000.

[Wolfe, 1994] Jeremy M. Wolfe. Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review*, 1(2):202–238, 1994.

Visual Analogy in Problem Solving

Jim Davies and Ashok K. Goel
College of Computing
Georgia Institute of Technology, Atlanta
jimmydavies@usa.net, goel@cc.gatech.edu

Abstract

Computational models of analogical problem solving have traditionally described source and target domains in terms of their causal structure. But psychological research shows that visual reasoning plays a part for many kinds of analogies. This paper describes a model that transfers a solution from a source analog to a new target problem using only visual knowledge represented symbolically. The knowledge representation is based on a language of primitive visual elements and transformations. We found that visual knowledge is sufficient for transfer, but that causal knowledge is needed to determine if the transferred solution is appropriate.

1 Introduction

The goal of this work is to examine the nature and role of visual representations and inferences in analogical reasoning, and especially in *analogical transfer*. Analogy involves learning about some *target analog* by transferring knowledge from a *source analog*. The process consists of several steps: *retrieval* is identifying a candidate source analog in memory; *mapping* is finding the best set of correspondences between components of the analogs; *transfer* is the application of knowledge from the source analog to the target analog; *evaluation* is determining if the target problem has been solved appropriately; *storage* is storing the target analog in memory for potential reuse.

Traditional conceptual and computational theories of analogy have focused primarily on causal knowledge and inferences (see [Holyoak and Thagard, 1997] [Bhatta and Goel, 1997] [Falkenhainer *et al.*, 1990] for examples). Psychological research, however, shows that visual reasoning often occurs in analogy [Holyoak and Thagard, 1997], [Pedone *et al.*, 1999] Some recent theories [Bhatta and Goel, 1997] [Griffith *et al.*, 2000] represent structural knowledge in addition to causal knowledge. Structural knowledge describes a system's physical composition but typically includes only the information directly relevant for analyzing the causal behaviors of the system. Structural knowledge might be thought of as a schematic that shows the components of the system and the connections among them but leaves out other visual information such as what a wire looks like, which side of a pump is up, etc.

We define visual representations as those that consist only of information relevant to how an image appears. Note that this definition of "visual" includes both high-level symbolic representations and low-level bitmap representations (which only represent the locations of points of light). We view visual and causal knowledge as lying on a spectrum, where one extreme has raw sensory data, (such as a *bitmap* image), and the other has highly interpreted and abstracted knowledge (e.g., teleological knowledge). Visual knowledge is closer to the perceptual, or *modal*, end of the spectrum, and causal knowledge is nearer to the *amodal* end. Causality can only be represented *implicitly* in a visual representation. In contrast to a bitmap image, the visual knowledge we use contains abstractions of objects and relations, and is thus represented symbolically.

Our hypothesis is that symbolically represented visual knowledge provides a level of abstraction at which two otherwise dissimilar domains may look more alike. For example, the concepts of an army on the march and a ray of radiation are quite different, but if both are represented as lines, it may facilitate analogical retrieval, mapping and transfer. We hypothesize that evaluation, on the other hand, requires explicit causal knowledge: simply because the path of the army and the ray look alike does not imply that the two behave similarly. Since other's work has begun to explore the use of visual knowledge for mapping, our work focuses on analogical transfer.

In this paper, we sketch an outline of our computational theory of visual analogical transfer for a class of problems in which the source analog contains a sequence of images (or can be analyzed in terms of an image sequence). This theory has been implemented in an operational computer program called Galatea. We illustrate the theory using the classical fortress/tumor problem [Duncker, 1926] as an example. This example was chosen because psychological data indicates that experimental participants used visual inferences in solving it [Holyoak and Thagard, 1997]. In this task, experimental participants first read a story about a problem-solving situation: A general with a large army wants to overthrow a dictator who lives in a fortress. All roads to the fortress are armed with mines that will go off if many people are on them at the same time. To solve this problem he breaks up his army into small groups and has them take different roads. The groups arrive at the same time and take the fortress. Then, the subjects are given a new problem: A patient needs radiation treatment on a tumor inside the body, but the radiation will harm the healthy tissue it reaches on the way in. Finally, the participants are asked to solve the tumor problem. The analogous solution is to target the tumor with low-level rays coming from different directions, and have them converge on the tumor.

2 Language and Processing

Our first task was to design a language to express visual analogs and the maps between them. Since the theory pertains to sequences of images, we needed both a vocabulary of primitive visual transformations that express changes between two consecutive images, and a vocabulary of primitive visual elements that enable the transformations. We designed a primitive visualization language, called *Privlan*, which consists of such primitive visual elements and primitive visual transformations. It can represent diagram-like images and changes to them. Like other computational visual analogy theories, ours represents images as networks of symbols. In Privlan symbolic images are called *simages*, to differentiate them from bitmap images.

Privels: Primitive Visual Elements

Each simage is composed of a collection of primitive visual elements, or *privels*. Table 1 shows a list of some privels.

Table 1

Privel name	attributes
generic-visual-element	location, size
line	start-point, end-point thickness, location
circle	location, size
box	location, height, width, orientation

Objects in the domain, like the fortress, are associated with a privel type. Each privel type has attributes associated with it. As shown in Table 1, *lines* have a *start-point*, an *end-point*, a *location* and a *thickness*. These attributes are not strongly-typed. For example, the *end-point* of a *line* could be a location such as the "center" of the image or some component of the image, like the fortress.

Privits: Primitive Visual Transformations

Privlan represents changes to images over time with an ordered series of simages in different states. Each simage in the sequence is connected to any simages before and after it with primitive visual transformations, or *privits*. Table 2 shows some examples of privits.

Table 2

Privit name	arguments
move	object, new-location
decompose	object, number-of-resultants
put-between	object, first-object, second-object
add-component	object

Each privit can take arguments. *Move*, for example, takes some *object* that it is moving, and a *new-location*. It changes the *object's* value for the *location* attribute to the *new-location*.

For example, imagine a circle moving from the top of the image to the bottom. Privlan would represent this as a series of two simages. The first simage would contain a *circle* with *location* set to top. The second would be to have another circle (called, say, *circle-1*) represented whose *location* would be set to bottom. Privlan knows these two simages are in a series because they are connected with a *transform-connection*, which in turn is associated with a series of correspondences between objects in the simages: There would be a map between the *circle* in the first simage and *circle-1* in the second. This map between the circles would be associated with the *move* privit.

2.1 Algorithm

The bottom series of simages in Figure 2 shows a representation of the solved fortress problem analog. The bottom left simage is the initial state of the problem. The top series of simages shows the target analog, the tumor problem. The darkly shaded box shows the output of the system. The first simage is all that is input of the tumor problem.

To make an analogical transfer, the source and target analogs must have an analogy between them. The analogy between the first tumor problem simage and the first fortress problem simage specifies maps between the components. To avoid over-complication of the figure, only one of these maps is shown, that between the *left-road1* and *left-body1*.

Privits are transferred from the bottom series to the top: *decompose* and *move*.

Following is the control structure for our visual analogical transfer theory. We will describe the transfer of the first privit as a running example. The process in the abstract can be seen in Figure 1.

1. **Identify the first simages of the target and analog problems.**
2. **Identify the privits and associated arguments in the current simage of the source analog.** This step finds out how the source problem gets from the current simage to the next simage. In our example, the privit is *decompose*, with "four" as the *number-of-resultants* argument (not shown).
3. **Identify the objects of the privits.** The object of the privit is what object the privit acts on. For the *decompose* privit is the *soldier-path1* (the thick arrow in the bottom left simage.)
4. **Identify the corresponding objects in the target analog.** The *ray1* (the thick arrow in the top left simage) is the corresponding component of the source analog's *soldier-path1*, as specified by the analogical map between the simages (not shown). A single object can be mapped to any number of other objects. If the object in question is mapped to more than one other object in the target, then the privit is applied to all of them in the next step. If the privit *arguments* are components of the

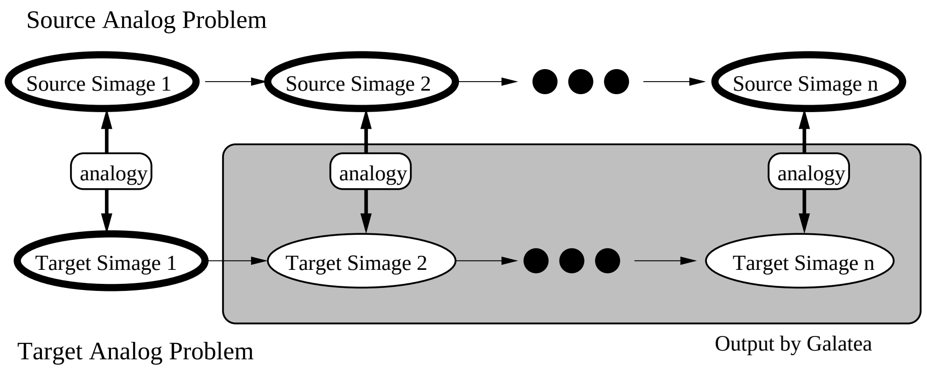

Figure 1: The things outside the shaded box are given to Galatea: a complete source problem an incomplete target problem, and the analogy between them. Galatea completes the analogical transfer and stores the new simage sequence for the target problem.

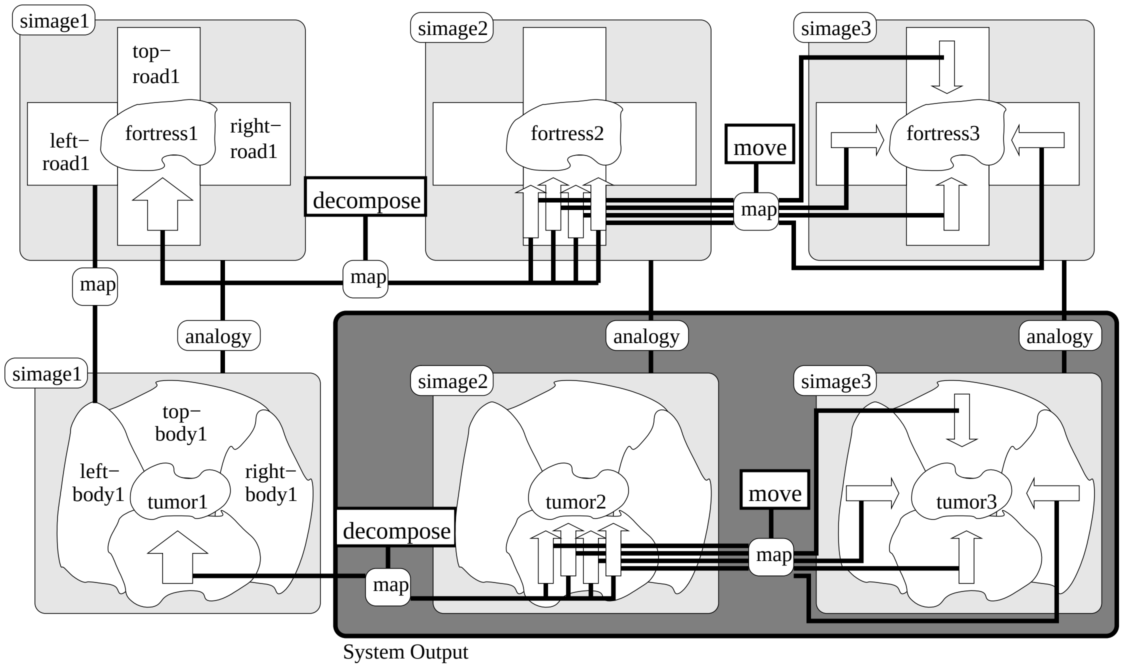

Figure 2: The fortress/tumor problem representation

source simage, then their analogs are found as well. Else the arguments are transferred literally.

5. **Apply the privit with the arguments to the target analog component.** A new simage is generated for the target problem (top middle) to record the effects of the privit. The *decompose* privit is applied to the *ray1*, with the argument "four." The result can be seen in the top middle simage in Figure 2. The new rays are created for this simage.

6. **Map the original objects to the new objects in the target problem.** A transform-connection and mapping are created between the target problem simage and the new simage (not shown). Maps are created between the corresponding objects. In this example it would mean a map between *ray1* in the first top simage and the four rays in the second top simage. The privit is associated with the map, as shown in the Figure, so the target problem itself can be used as a possible source analog in the future.

7. **Map the new objects of the target problem to the corresponding objects in the source problem.** In this case the rays of the second target simage are mapped to soldier paths in the second source simage. This step is necessary for the later iterations (i.e. going on to another transformation and simage). Otherwise the system would have no way of knowing which parts of the target simage the later privits would operate on.

8. **Check to see if goal conditions are satisfied.** If they are, exit, and the problem is solved. If not, and there are further simages in the source series, set the current simage equal to the next simage and go to step 1. If there are no further simages, then exit and fail.

3 System: Galatea

Our hypothesis was that a visual representation language would be sufficient to describe domains such that analogical problem solving could take place. To test this hypothesis, we implemented the above ideas in a program called Galatea, and applied it to Duncker's fortress/tumor analogy.

Galatea's knowledge representation architecture consists of two kinds of propositions: 1. A statement of existence of a concept or relation and 2. The connection of two concepts or propositions with a relation.

Galatea takes as input a solved source problem, an unsolved target problem (both represented visually), an analogical mappings between the simages, and criteria for an adequate problem solution. When instructed to solve the target using the source, it analogically transfers the solution procedure. As can be seen in Figure 2, it outputs a series of simages for the target problem, and checks to see if the solution transferred indeed solves the problem constraints. The following section describes our results.

Duncker's Fortress/Tumor Problem

Table 3 shows some of the privels and their attribute values for the first fortress problem simage.

Table 3: Privels from Fortress Problem Simage 1

Visual Object	attributes	value
Fortress	looks-like:	generic-visual-element
	location:	center
Bottom-road	looks-like:	line
	start-point:	bottom
	end-point:	fortress
Right-road	looks-like:	line
	start-point:	right
	end-point:	fortress
Left-road	looks-like:	line
	start-point:	left
	end-point:	fortress
Top-road	looks-like:	line
	start-point:	top
	end-point:	fortress
Soldier-path	looks-like:	line
	location:	bottom-road
	thickness:	thick

We represented the fortress story with three simages (see Figure 2.) The first was a representation of the original fortress problem. It had four roads, represented as thick lines, radiating out from the fortress, which was a *generic-visual-element* in the center. We represented the original soldier path as a thick line on the bottom road. This simage was connected to the second with a *decompose* privit, where the arguments were *soldier-path1* for the *object* and "four" for the *number-of-resultants*. The second simage shows the *soldier-path1* decomposed into four thin lines, all still on the bottom road. The lines are thinner to represent smaller groups. This is connected to the final simage with the *move* privit, which is applied to three of the new soldier paths. They are sent to the different roads. The final simage in the fortress problem shows all four soldier paths, each on a different road.

We represented the start state of the tumor problem as a single simage. The tumor itself is represented as a *generic-visual-element*. The *ray* of radiation is a thick *line* that passes through the bottom body part.

Galatea transfers the first transformation (*decompose*) from the source analog (the solved fortress problem) to the target (the tumor problem). It knows which part of the tumor problem to apply this privit to from the given analogical mapping between the first simages of the fortress and tumor problems. Galatea generates a second simage with the *line* representing the ray decomposed into four thinner lines. In the next iteration Galatea successfully transfers the second transformation, moving each of the rays to the different roads.

Galatea can solve analogical transfer problems using only visual knowledge, as we have shown with the fortress/radiation example. Though this work is still in progress, we conjecture that this theory, when Privlan is more fleshed out, will apply to all problems whose solution constraints involve visually perceivable states of the world. Another sense of this is: if you can make a diagram of it, our theory applies to it.

3.1 Causal Knowledge

Though the solution procedure was transferred in both of these cases, the system still had no way of knowing if the transferred solution was *adequate* for the new problem. In the tumor problem, in order for the agent to determine if the tumor was destroyed and the patient was still alive, it needed some causal knowledge. By causal we mean knowledge of how things in a system change as they interact. Pre- and post-conditions are a straightforward way to represent this, but it is difficult to imagine what "visual" pre- and post-conditions might look like. Visual representations alone cannot enable evaluation of the solution.

Galatea represents causal knowledge with production rules, implemented in ACT-R [Anderson and Libiere, 1998]. We have no theoretical commitment to production rules or ACT-R. One production rule identifies a body part as dead if there is a thick line representing a ray going through it. Another rule identifies the tumor being killed if enough radiation is hitting it. If the tumor is dead and the body is alive, a final production fires that identifies the problem as being solved.

When the tumor problem is first encountered (when it only consists of a single simage), Galatea is unable to infer through the productions that the problem is solved in the initial state. When the solution is transferred from the fortress, the rules confirm that the problem has been solved.

4 Discussion

In our earlier work, we have developed a theory of Model-Based Analogy based on Structure-Behavior-Function models of causal mechanisms and physical systems. The IDeAL system [Bhatta and Goel, 1997], for example, transfers generic teleological mechanisms from a source analog to a target problem to address novel design problems. The ToRQUE system [Griffith *et al.*, 2000] uses generic structural transformations to mutate a target problem or a source analog to construct analogies. Galatea builds on the above theory of model-based analogy in that in it too relies on the core idea of generic transformations. Thus, while the analogical process in Galatea is similar to that in IDeAL, the content of its generic transformations is visual as opposed to teleological or structural. ToRQUE's structural knowledge captures only a small subset of visual knowledge. In contrast Galatea has information about the location and appearance of objects in a particular simage: the fortress is not just connected to the road, it is in the center of the simage; the path is not just on the road, it is a thick line. These additional features enable the initial analogical mapping between simages without causal knowledge because the simages representing the two analogs are similar when described visually.

Like Galatea, LetterSpirit is a model of analogical transfer [McGraw and Hofstadter, 1993]. It takes a stylized seed letter as input and outputs an entire font that has the same style. It does this by determining what letter is presented, determining how the components are drawn, and then drawing the same components of other letters the same way. Like Galatea, the analogies between letters are already in the system: the vertical bar part of the letter "d" maps to the vertical bar in the letter "b," for example. A mapping is created for the input character. For example, the seed letter may be interpreted as an "f" with the cross-bar suppressed. When the system makes a lower-case "t," by analogy, it suppresses the crossbar.

It is not at all clear that LetterSpirit is applicable to other domains (such as the fortress/tumor problem) in part because there is little distinction between its theory and the implementation that works for letters. In contrast, one can see how Galatea might be applied to the font domain: The stylistic guidelines in LetterSpirit, such as "crossbar suppressed" are like the visual transformations in Galatea: it would be a transformation of removing an element from the image, where that element was the crossbar and the image was a prototype letter "f." Then the transformation could be applied to the other letters one by one. We conjecture that our theory has more generality than LetterSpirit.

Galatea does not generate the analogical mapping, but other systems, that create mappings with visual information, have shown that it can be done. The VAMP systems are analogical mappers as well [Thagard *et al.*, 1992]. VAMP.1 uses a hierarchically organized symbol/pixel representation. It superimposes two images, and reports which components have overlapping pixels. VAMP.2 represented images as agents with local knowledge. Mapping is done using ACME/ARCS [Holyoak and Thagard, 1997], a constraint satisfaction connectionist network. The radiation problem mapping was one of the examples to which VAMP.2 was applied.

The Structure Mapping Engine, or SME [Falkenhainer *et al.*, 1990] finds the best mapping of elements between two domains. But SME typically is applied to instances where the situations are represented as having causal and structural knowledge. SME has been applied to visual knowledge in a system called MAGI [Ferguson, 1994], which takes visual representations and uses SME to find examples of symmetry and repetition in a single image.

Like Galatea, MAGI and the VAMPs use visual knowledge. But unlike Galatea their focus is on the creation of the mapping rather than on transfer of a solution procedure. MAGI's and Galatea's theories are compatible: a MAGI-like system might be used to create the mappings that Galatea uses to transfer knowledge. The theory behind the VAMPs is incompatible because they use a different level of representation for the images.

Galatea has also been applied to the case study of James Clerk Maxwell's creation of his electro magnetic theory. According to Nersessian's Cognitive-Historical Analysis [Nersessian, 1995], Maxwell used analogical transfer to resolve a problem with his mental model of electro-magnetism. The transfer was mediated by a generic abstraction, and the abstraction was created, retrieved, and instantiated using visual representations and reasoning.

Galatea so far has been substantiated for only two examples: Duncker's fortress/tumor problem and Maxwell's case study. In the future, we will extend Galatea to cover many more problems, and expand it to use, in addition to simages, bitmap images, which we believe will be important for changing representations when symbol mismatches make analogical mapping difficult.

5 Conclusion

The first finding of our experiments with Galatea is that visual knowledge alone, with no explicit representation of causal knowledge, is sufficient for enabling analogical transfer. This validates the central hypothesis of our work. Galatea suggests a computational model of analogy based on dynamic visual knowledge that complements traditional models based on causal knowledge. Although Galatea does not address the issues of retrieval and mapping, put together with other work described in the previous section, we can now more confidently conjecture that visual knowledge alone can enable retrieval, mapping and transfer in analogy.

A second finding of our work on Galatea is that evaluation, in general, cannot be done using visual knowledge alone; it requires causal knowledge too. Thus visual knowledge enables only the steps that depend directly on the visual similarity between the target problem and the source analog, e.g., retrieval, mapping and transfer. It does not, however, fully support the evaluation step because it depends not on similarity but on the intrinsic causal and teleological structure of the target problem.

Galatea represents visual knowledge symbolically, in the form of symbolic images made of primitive visual elements and primitive visual transformations. The symbolic representation provides the standard benefits of discreteness, abstraction, ordering, and composition. Although sequences of lower-level *bitmap* representations also capture the notion of ordering, they, by themselves, neither capture abstractions that enable noticing visual similarity nor enable transformations on the images. This leads us to a third finding: Galatea provides additional evidence that symbolic representations of visual images are necessary for analogy.

6 Acknowledgements

This work has benefitted from many discussions with Nancy J. Nersessian at the Georgia Institute of Technology. This research has been inspired in part by the work of B. Chandrasekaran at the Ohio State University and Lawrence Barsalou at Emory University in addition to that of Nersessian. Goel also benefited from discussions with David Waltz at NECI during his 1998 sabbatical.

References

[Anderson and Libiere, 1998] John R. Anderson and Christian Libiere. *The Atomic Components of Thought.* Lawrence Erlbaum Associates, 1998.

[Bhatta and Goel, 1997] Sambasiva R. Bhatta and Ashok K. Goel. A functional theory of design patterns. In *Proceedings of Fifteenth International Joint Conference on Artificial Intelligence*, pages 294–300, Nagoya, Japan, August 1997.

[Duncker, 1926] K. Duncker (1926). A qualitative (experimental and theoretical) study of productive thinking (solving of comprehensible problems). *Journal of Genetic Psychology*, 33:642–708, 1926.

[Falkenhainer *et al.*, 1990] Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The Structure mapping engine: algorithm and examples. *Artificial Intelligence*, 41:1-63, 1990.

[Ferguson, 1994] Ronald W. Ferguson. MAGI: Analogy-based encoding using regularity and symmetry. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* pages 283-288. Atlanta, Georgia: Lawrence Erlbaum Associates, 1994.

[Griffith *et al.*, 2000] Todd W. Griffith, Nancy J. Nersessian and Ashok K. Goel. Function-follows-form transformations in scientific problem solving. In *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2000.

[Holyoak and Thagard, 1997] Keith J. Holyoak and Paul Thagard. The analogical mind. *American Psychologist*, 52(1):35–44, 1997.

[Kosslyn, 1994] Kosslyn, S. M. *Image and Brain: The Resolution of the Imagery Debate.* MIT Press, Cambridge, MA, 1994.

[McGraw and Hofstadter, 1993] G. McGraw and Douglas R. Hofstadter. Perception and creation of alphabetic style. In *Artificial Intelligence and Creativity: Papers from the 1993 Spring Symposium*, AAAI Technical Report SS-93-01, AAAI Press, 1993.

[Nersessian, 1995] Nersessian, N. J. Opening the black box: Cognitive science and the history of science. In Thackray, A. (ed.) *Constructing Knowledge in the History of Science. Osiris* 10, 1995.

[Pedone *et al.*, 1999] R. Pedone, John E. Hummel, and Keith J. Holyoak. The use of diagrams in analogical problem solving. Manuscript draft, 1999.

[Thagard *et al.*, 1992] Paul Thagard, D. Gochfeld, and S. Hardy. Visual analogical mapping. In *Proceedings of the 14th Annual Conference of the Cognitive Science Society*, pages 522–527, Hillsdale, Erlbaum, 1992.

COGNITIVE MODELING

COGNITIVE MODELING — CATEGORIZATION

Reasoning about Categories in Conceptual Spaces

Peter Gärdenfors
Lund University Cognitive Science
Lund University
S-222 22 Lund, Sweden
peter.gardenfors@lucs.lu.se

Mary-Anne Williams
Business and Technology Research Lab
University of Newcastle
NSW 2308, Australia
maryanne@cafe.newcastle.edu.au

Abstract

Understanding the process of categorization is a primary research goal in artificial intelligence. The conceptual space framework provides a flexible approach to modeling context-sensitive categorization via a geometrical representation designed for modeling and managing concepts.

In this paper we show how algorithms developed in computational geometry, and the Region Connection Calculus can be used to model important aspects of categorization in conceptual spaces. In particular, we demonstrate the feasibility of using existing geometric algorithms to build and manage categories in conceptual spaces, and we show how the Region Connection Calculus can be used to reason about categories and other conceptual regions.

1 Introduction

Categorization is a fundamental cognitive activity. The ability to classify and identify objects with a high degree of exception tolerance is a hallmark of intelligence, and an essential skill for learning and communication. Understanding the processes involved in constructing categories is a primary research goal in artificial intelligence.

The conceptual space framework as developed by Gärdenfors [2000] provides a flexible approach to modeling context-sensitive categorization. Conceptual spaces are based on a simple, yet powerful, geometrical representation designed for modeling and managing concepts.

In this paper we show how algorithms developed in computational geometry, and the Region Connection Calculus (RCC) [Cohn *et al.*, 1997], a well known region-based spatial reasoning framework, can be used to model important aspects of categorization in conceptual spaces. In particular, we demonstrate the feasibility of using existing geometric algorithms to build and manage categories in conceptual spaces, and we show how the RCC can be used to reason about categories and other conceptual regions.

2 Conceptual Spaces

Conceptual spaces provide a framework for modeling the formation and the evolution of concepts. They can be used to explain psychological phenomena, and to design intelligent agents [Chella *et al.*, 1998; Gärdenfors, 2000]. For the purposes of this paper conceptual spaces provide the necessary infrastructure for modeling the process of categorization.

Conceptual spaces are geometrical structures based on quality dimensions. Quality dimensions correspond to the ways in which stimuli are judged to be similar or different. Judgments of similarity and difference typically generate an ordering relation of stimuli, e.g. judgments of pitch generate a natural ordering from “low” to “high” [Gärdenfors, 2000]. There have been extensive studies conducted over the years that have explored psychological similarity judgments by exposing human subjects to various physical stimuli. Multi-dimensional scaling is a standard technique that can be used to transform similarity judgments into a conceptual space [Krusal and Wish, 1978]. An interesting line of inquiry is pursued by Balkenius [1999] who attempts to explain how quality dimensions in conceptual spaces could accrue from psychobiological activity in the brain.

In conceptual spaces objects are characterized by a set of attributes or qualities $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_n\}$. Each quality $\mathbf{q}_i$ takes values in a domain $\mathbf{Q}_i$. For example, the quality of pitch (or frequency) for musical tones could take values in the domain of positive real numbers. Objects are identified with points in the conceptual space $\mathbf{C} = \mathbf{Q}_1 \text{ x } \mathbf{Q}_2 \text{ x } \ldots \mathbf{Q}_n$, and concepts are regions in conceptual space.

In the definition above we use the standard mathematical interpretation of “domain”. In [Gärdenfors, 2000] however, a domain is defined to be a set of *integral dimensions*, this interpretation is consistent with its use in the psychology literature. For example, pitch and volume constitute the integral dimensions of sounds discernible by the human auditory perception system. Integral dimensions are such that they cannot be separated in the perceptual sense. The ability to bundle up integral dimensions as a domain is an important part of the conceptual spaces framework. Domains facilitate the sharing and inheritance of integral dimensions across conceptual spaces.

For the purpose of this paper, and without loss of generality, we often identify a conceptual space $\mathbf{C}$ with $\mathbf{R}^n$, but hasten to note that conceptual spaces do not require the full richness of $\mathbf{R}^n$. Domains can be continuous or discrete[1]. They

[1] They can even be small and finite e.g. {male, female}.

can also be based on a wide range of geometrical structures, for example, according to psychological evidence the human colour perception system is best represented using polar coordinates[2] [Gärdenfors, 2000].

For the purpose of problem solving, learning and communication, agents adopt a range of conceptualizations using different conceptual spaces depending on the cognitive task at hand.

Similarity relations are fundamental to conceptual spaces. They capture information about the similarity judgments. In order to model some similarity relations we can endow a conceptual space with a distance measure.

Definition 1 *A* distance measure d *is a function from* **C** x **C** *into* **T** *where* **C** *is a conceptual space and* **T** *is a totally ordered set.*

Distance measures lead to a natural model of similarity; the smaller the distance between two objects in conceptual space, the more similar they are. The relationship between distance and similarity need not be linear, e.g. similarity may decay exponentially with distance.

The properties of connectedness, star-shapedness and convexity of regions in conceptual spaces will prove useful throughout.

Definition 2 *A subset C of a conceptual space is:*

(i) connected *if for every decomposition into the sum of two nonempty sets $C = C_1 \cup C_2$, we have $\bar{C}_1 \cap C_2 \cup C_1 \cap \bar{C}_2 \neq \emptyset$ where $\bar{C}$ is the closure of C. In other words, C is connected if it is not the disjoint union of two non-empty closed sets.*

(ii) star-shaped with respect to a point **p** *(referred to as a* kernel point*) if, for all points* **x** *in C, all points between* **x** *and* **p** *are also in C.*

(iii) convex *if, for all points* **x** *and* **y** *in C, all points between* **x** *and* **y** *are also in C.*

Definition 3 *The* kernel of a star-shaped region C *is the set of all possible kernel points, and will be denoted $kernel(C)$.*

Connectedness is a topological notion, whilst star-shapedness and convexity rely only on a betweenness relation. A qualitative betweenness relation can be specified in terms of a similarity relation, $S(\mathbf{a}, \mathbf{b}, \mathbf{c})$, which says that **a** is more similar **b** than it is to **c**. Alternatively, a betweenness relation can be used as primitive, and axioms introduced to govern its behaviour [Borsuk and Szmielew, 1960]. In the special case where the distance measure is a metric, the betweenness relation can be defined as: "**b** is between **a** and **c**" if and only if $d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c}) = d(\mathbf{a}, \mathbf{c})$.

Convex regions are star-shaped, and in many topological settings star-shaped regions are connected. The kernel of a convex region is the region itself, and under the Euclidean metric kernels are convex.

[2]A scientific representation of colour would require a different representation however, one that captures important scientific features of the electromagnetic spectrum such that the wave properties of wavelength and amplitude constitute integral dimensions.

Constraints like connectedness, star-shapedness and convexity can be used to impose ontological structure on the categorization of the conceptual space, i.e. not any old region can serve as a category. In fact, there is compelling evidence that *natural properties* correspond to convex regions in conceptual space, and using the idea of a natural property in this way Gärdenfors [2000] is able to sidestep the enigmatic problems associated with induction.

In section 4 we show how categorization, the central theme of this paper, occurs in conceptual spaces, but first we briefly describe the RCC.

3 Region Connection Calculus

The RCC is a qualitative approach to spatial reasoning. It was developed in an attempt to build a commonsense reasoning model for space, and its remarkable utility has been illustrated in numerous innovative applications [Cohn *et al.*, 1997].

The RCC approach is region-based where spatial regions are identified with their closures. The RCC is based on a connection relation, $C(X, Y)$, which stands for "region X connects with region Y". The connection relation, C, is reflexive and symmetric. Despite the fact that the basic building blocks in the RCC are regions, C can be given a topological interpretation, namely $C(X, Y)$ holds when the topological closures of regions X and Y share at least one point.

The RCC framework comprises several families of binary topological relations. One family, the RCC5 fragment uses the following Jointly Exhaustive and Pairwise Disjoint base relations to describe the relationship between two regions (see Figure 1); DR (discrete), EQ (identical), PP (proper part), PP^{-1}(inverse PP), and PO (partial overlap).

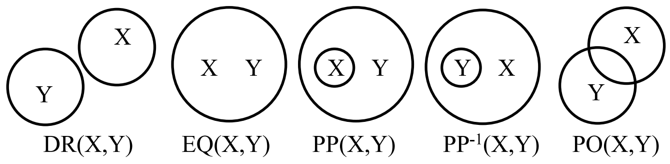

Figure 1: *The base relations in RCC5.*

Boundaries of regions are not distinguished in RCC5; there is no difference between two regions being totally disconnected and externally connected, and no difference between a proper part tangentially connected to the boundary and a proper part disconnected from the boundary. Another fragment, RCC8, possesses base relations that can make these distinctions.

In this paper our interests lie in similarity based categorization, and we use RCC5 to illustrate how the RCC can be used to represent conceptual regions. Extending to a more expressive mereotopological language to reason about the adjacency of categories, for example, is straightforward, and it can be done at no extra computational cost.

Transition tables can be used to perform reasoning about relationships between regions in RCC5 and RCC8. It is known that there is no complete first order finite axiomatization of topology. Nebel [1995] showed that propositional reasoning in RCC5 and RCC8 is NP-Hard, and Renz and Nebel

[1999] identified a maximal tractable subset of relations that contain all the base relations. Efficient implementations in Prolog have been constructed using a zero-order intuitionistic logic [Bennett, 1997].

The RCC can be used to represent spatial regions and to reason about them in any dimension, provided that all spatial entities possess precisely the same number of dimensions. It can also represent regions composed of multiple (spatially distinct) parts, but it cannot represent null regions.

In this paper we use the RCC to reason about spatial relationships between conceptual regions, and to model the indeterminacy of conceptual regions. The original formulation of the RCC concerned regions with crisp well-defined boundaries, but later Cohn and Gotts [1996] extended the RCC to handle indeterminate vague regions that could be "crisped" into less vague regions. We describe this crisping relation in more detail in section 5.2.

4 Categorization in Conceptual Spaces

4.1 Prototypes and Categorical Perception

A categorization results in a partitioning of a conceptual space into (meaningful) subregions. The geometrical nature of conceptual spaces coupled with representations for prototypes and the ability to manipulate dimensions independently of one another ensures that they provide a highly flexible and practical representation of context-sensitive categorization.

Within each category certain members are judged to be more representative than others [Rosch, 1975]. The most representative members of a category are *prototypes*. There is a wealth of psychological data supporting the existence of prototypes and their key role in categorization. Typically human performance experiments are used to determine how well, and how quickly humans can classify, label, rank or compare objects. Experimental results consistently show that the ease of classification varies with how similar an object is to a prototype. Furthermore, the more similar a nonmember is to the prototype, the more difficult it is to exclude. It has been shown, for example, that the reaction time recognizing that a robin is a bird is shorter than identifying a penguin as a bird regardless of whether the stimulus is the name or an image.

It is evident from experiments that humans make judgments about the degree of resemblance to a prototype during classification and identification; a robin is judged as a more prototypical bird than a penguin [Harnad, 1987]. A prototype consists of features of either a typical, or ideal category member, rather than invariant features common to every member. For example, a prototype of a category can be thought of as an amalgam of the characteristic attributes of its category's exemplars[3].

Classifying an object using prototypes is accomplished by determining its similarity to a prototype. Instances above some *threshold* of similarity to the prototype are taken as category members, all other instances are nonmembers.

Prototypes are central to the representation and processing of categories. People classify, generate, acquire, and reason about typical exemplars faster and more accurately than atypical exemplars. They also produce stronger inductive inferences with typical exemplars than with atypical exemplars [Hampton, 1993].

It is widely accepted on the basis of empirical psychological experiments that people sometimes judge membership of categories as graded. The existence of graded concepts supports the notion of *continuous perception*. The gradedness of category membership can be used to determine how closely an object resembles a prototype and can naturally be determined from the underlying similarity judgments.

Psychological evidence also suggests that people distinguish stimuli along a physical continuum much better when the stimuli are from different categories than when they are from the same category. This phenomenon is called *categorical perception* [Harnad, 1987], and is manifested in the ability to discriminate stimuli with more ease and accuracy between categories than within them. When categorical perception is at work, stimuli related to a specific category are perceived as indistinguishable, whereas stimuli from a "nearby"category are perceived to be entirely different. This phenomenon has been found in the way humans process sounds in speech. In color perception, for example, different shades of green are perceived to be more similar than green and yellow even though the wavelength differences are no larger. In other words, the psychophysical relationship between the physical intensity of a stimulus and the psychological intensity of the ensuing sensation is related to the categorization. Categorical perception has also be found in primates, other than humans [May *et al.*, 1989].

It is also well known that similarity judgments crucially depend on the context in which they occur for both continuous and categorical perception. It turns that certain features of objects and concepts are more salient for a particular categorization (for both classification and identification) depending on the context. The classic example is that a robin is a prototypical bird, but a canary is a prototypical pet bird.

In summary the key findings from psychological studies of categorization are (i) similarity judgments play a fundamental role in categorization and they are context sensitive, (ii) the degree of similarity is judged with respect to a reference object/region such as a prototype, (iii) category membership can be graded (discrete membership, if and when it exists, is considered to be a special case), and (iv) the psychophysical relationship between the stimulus and the response depends on the underlying categorization.

4.2 A Mechanism for Building Categories

In this section results in computational geometry are applied to categorization in conceptual spaces. We provide computational evidence for categorization based on prototypes, rather than appealing to the usual intuitive arguments found in the psychology literature derived from facts like the presentation of prototypes enhances learning. It can be shown, for example, that the conceptual space model predicts that it is easier to learn categories in which the natural prototype is central to a set of variations than it is to learn categories in which the prototype occurs as a peripheral member, as observed by Rosch [1975].

[3]Exemplars are previously perceived examples of objects in a category.

The main idea is that Voronoi tessellations around prototypes can be used to determine the threshold of similarity that forms category boundaries. In other words, the prototypes and the underlying similarity relation can be used to tessellate a conceptual space into categories.

Definition 4 *A Voronoi tessellation in conceptual space is given by the triple* $\Delta(\mathbf{P}, d, \mathbf{C})$ *where* $\mathbf{P}$ *is a set of distinct generator points* $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m\}$, d *is a distance measure on* $\mathbf{C}$ *a conceptual space. We define the tessellated regions* $c(\mathbf{p}_i)$ *to be* $\{\mathbf{x} | d(\mathbf{p}_i, \mathbf{x}) \leq d(\mathbf{p}_j, \mathbf{x})$ *for* $j = 1, 2, \ldots, m\}$*, and we call* $c(\mathbf{p}_i)$ *the category generated by* $\mathbf{p}_i$.

A Voronoi tessellation divides a conceptual space according to the *nearest-neighbor rule* which says each point/object in the space is associated with the prototype closest to it. This results in prototypes being centrally located in their category.

The Euclidean metric is a basic distance measure: the Euclidean distance $d(\mathbf{x}, \mathbf{p})$ between two points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ in $\mathbf{R}^n$ is calculated as $\sqrt{\sum_{j=1}^{n}(x_j - p_j)^2}$. If the underlying distance measure is the Euclidean metric, then the resultant categories $c(\mathbf{p}_i)$, are convex, and hence star-shaped with respect to $\mathbf{p}_i$. If the distance measure is the Manhattan or the supremum metric, then the generated categories are not necessarily convex, but are star-shaped with respect to $\mathbf{p}_i$. In fact, it can be shown that if $\mathbf{x}$ is between $\mathbf{y}$ and $\mathbf{p}_i$ is defined as $d(\mathbf{p}_i, \mathbf{x}) + d(\mathbf{x}, \mathbf{y}) = d(\mathbf{p}_i, \mathbf{y})$ and the distance measure satisfies the triangle inequality, then the generated categories are star-shaped with respect to $\mathbf{p}_i$.

Star-shapedness with respect to a prototype $\mathbf{p}$ is a desirable property for categories: if a category $c(\mathbf{p})$ is not star-shaped with respect to $\mathbf{p}$, then there is an object $\mathbf{x}$ that is between $\mathbf{p}$ and some $\mathbf{y} \in c(\mathbf{p})$ but $\mathbf{x} \notin c(\mathbf{p})$.

Definition 5 *A* well behaved categorization *in conceptual space produces regions which are star-shaped with respect to their prototype region and contain their central prototype.*

A Voronoi tessellation encapsulates the entire proximity information about the set of prototypes in a computationally compact fashion. Voronoi diagrams in the plane can be computed in $O(n \ log \ n)$ worst-case optimal time using $O(n)$ space [Okabe $et \ al$, 2000], and in d-dimensions for $d > 3$ in $O(n^{\lceil d/2 \rceil})$ worst-case optimal time [Klee, 1980].

Once constructed Voronoi tessellations can be used to: (i) identify the category of arbitrary objects in logarithmic query time without increasing the storage space - this is asymptotically optimal since it matches the information theoretical lower bound [Auberhammer, 1991], and (ii) compute the smallest enclosing sphere containing n prototype points in $O(n \ log \ n)$ worst-case optimal time [Auberhammer, 1987]. Furthermore, a prototype can be added or deleted to a Voronoi tessellation in $O(n)$ time, and two Voronoi tessellations can be merged in $O(n)$ time.

It is interesting to note that classical techniques and algorithms for information retrieval and cluster analysis are related to Voronoi tessellations. In fact, Voronoi tessellations have been used to construct robust approximate solutions to well known NP-complete information retrieval problems, i.e. acceptable approximate solutions can be found in $O(n \ log \ n)$ time [Okabe $et \ al$, 2000].

Much experimental psychological data concords with the idea of tessellating conceptual spaces into star-shaped (and sometimes convex) regions around prototypes or exemplars, e.g. stop consonants in phoneme classification [Petitot, 1989], other examples can be found in [Gärdenfors, 2000].

Not only do Voronoi tessellations generated by prototypes support the prototype model of categorization, but the generated boundaries provide a threshold of similarity and support a mechanism which can explain categorical perception. The precise mechanism involves crisping the distance measure and is described in Section 5.2.

4.3 Generalized Voronoi Tessellations

In this section we discuss some useful extensions of the basic Voronoi tessellation model. Ordinary Voronoi tessellations give rise to ideal categorizations, in the sense that crisp boundaries are generated from a single prototype. We generalize the definition so as to generate categories from conceptual regions rather than specific prototype points.

Definition 6 *A generalized Voronoi tessellation is given by the triple* $\Delta(\mathbf{P}, d, \mathbf{C})$ *where* $\mathbf{P}$ *is a set of generator regions* $\{P_1, P_2, \ldots, P_m\}$, d *is a distance measure on* $\mathbf{C}$ *a conceptual space. We define the tessellated regions* $c(P_i)$ *to be* $\{\mathbf{x} | d(P_i, \mathbf{x}) \leq d(P_j, \mathbf{x})$ *for* $j = 1, 2, \ldots, m\}$*, and we call* $c(P_i)$ *the category generated by the region* P_i.

We contrast two distance measures for generalized Voronoi categorizations; the *additively weighted distance* and the *power distance*[4]. The additively weighted Voronoi diagram is typically used to model the growth of biological cells, and can be used to model the growth of concepts also. The power distance, on the other hand, is best suited to handle indeterminacy and exemplar variability.

An additively weighted distance between a point $\mathbf{x}$ and a sphere $P \in \mathbf{P}$ in $\mathbf{R}^n$ with weight $w(P)$, denoted $d(\mathbf{x}, P)$, is defined as $d(\mathbf{x}, \mathbf{p}) - w(P))$ where $d(\mathbf{x}, \mathbf{p})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{p}$ the center of P. A common way to define the additively weighted distance between a point $\mathbf{x}$ and a sphere P is to take $w(P)$ as the radius r_p of P, i.e. $d(\mathbf{x}, P) = d(\mathbf{x}, \mathbf{p}) - r_p$, which can naturally be interpreted as the shortest distance between the point $\mathbf{x}$ and the surface of the sphere P, see Figure 2(a). The resulting tessellation is called the Euclidean weighted Voronoi diagram. A point $\mathbf{x}$ lies on or inside the sphere P if and only if $d(\mathbf{x}, P) \leq 0$. Okabe *et al.* [2000] proved that the bisector of a Euclidean weighted Voronoi diagram is either a hyperbolic surface or a hyperplane, and that the generated regions are connected and star-shaped with respect to its generator sphere in $\mathbf{R}^n$. This result also holds for the Manhattan and the supremum metrics.

One way to obtain convex regions (with straight line bisectors) is to use the power distance (also known as the Laguerre distance): $d(\mathbf{x}, P) = \sqrt{d(\mathbf{x}, \mathbf{p})^2 - r_p^2}$. When $\mathbf{x}$ is outside the sphere P centered on $\mathbf{p}$ the distance from P to $\mathbf{x}$ is given by the length of the tangent from P to $\mathbf{x}$. The power distance and power bisector are illustrated in Figure 2(b) and (c), respectively.

[4]A related, but different, measure is used by Gärdenfors [2000].

The size of a particular prototype's radius relative to the surrounding prototype regions reflects its ability to influence its neighborhood. The magnitude of the radius can be related to the actual size of the category, the variability among the exemplars[5], or the correlation of qualities.

Definition 7 *We define a* power categorization *to be a generalized Voronoi tessellation* $\Delta(\mathbf{P}, d, \mathbf{C})$ *generated by a set of prototype regions* $\mathbf{P}$ *using the power distance.*

If the radii of the generator spheres are zero or equal in size, then the power categorization will be equivalent to the categorization based in the ordinary (point-based) Euclidean Voronoi tessellation.

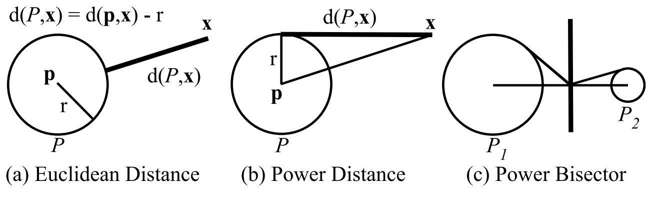

Figure 2: *The Euclidean and the power distance measures.*

The worst-case time complexity for the construction of the power categorization is no worse than that for the ordinary Voronoi diagram. Sometimes the structure of the space can be exploited, which means that the actual computational time can be dramatically lowered. Voronoi tessellations are used in a wide range of applications and domain constraints can be used to improve algorithms, typically linear time can be expected, e.g. linear time can be expected if the generating spheres/points are uniformly distributed [Dwyer, 1987].

Parallel algorithms have also been developed [Auberhammer 1991] which construct Voronoi diagrams in $O(log\ n)$ time using $O(n)$ processors.

It turns out that for power categorizations, if a generator sphere P_1 is a proper part of another generator sphere P_2 then the generated category $c(P_1)$ will not contain the center of the generator region P_1, and a well-behaved categorization will not be produced. For example, using the generic $bird$ and the $robin$ prototype regions in Figure 3 to generate a power categorization would not result in a well behaved categorization; since $PP(robin, bird)$, it turns out that $robin \notin c(robin)$ in the power categorization.

RCC5 can be used to ensure that generating spheres are not proper parts of other generating spheres, and hence can play a role in the categorization process itself, by determining the legitimate spheres to use as generators.

Finally, we define the notion of a bounded tessellation which provides a useful mechanism for selecting conceptual regions to focus on for conceptual spatial reasoning and categorization.

[5]The standard deviation of the exemplars from the prototype could be used [Gärdenfors, 2000]. In the bird conceptual space the standard deviation of birds is larger than that of emus, so we might expect the sphere that generates the bird category to be larger than that used for emus as in Figure 3.

Definition 8 *Given a generalized Voronoi tessellation* $\Delta(\mathbf{P}, d, \mathbf{C})$ *where the categories are generated by* $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ *define the Voronoi tessellation bounded by a region* S, *denoted by* $\mathbf{C}_{\cap S}$, *to be* $\{c(\mathbf{p}_1) \cap S, c(\mathbf{p}_2) \cap S, ..., c(\mathbf{p}_n) \cap S\}$. *We denote the bounded categories* $c(\mathbf{p}_i) \cap S$ *by* $c_{\cap S}(\mathbf{p}_i)$.

A bounded Voronoi diagram may be disconnected if every boundary region is not star-shaped with respect to its generator point [Okabe *et al.*, 2000].

5 Reasoning about Categories

Concept management involves categorization, concept acquisition, concept formation and conceptual change. Cognitive processes such as learning and communication impel and guide concept management. In the previous section we showed how conceptual spaces provide a rich and computationally effective representation for categorization based on prototype regions, and in this section we show that the RCC machinery can be used to reason about categories and to describe other aspects of concept management.

5.1 Determining Spatial Relationships

The RCC can be used to determine the relative configuration of conceptual regions such as categories, concepts, prototypes, and exemplars. For example we can determine: (i) if the smallest region containing all the prototypes is a proper part of a given category, (ii) if some category overlaps another category e.g. $PP(c(robin), c(bird))$, (iii) if the region containing all the prototypes contains all the exemplars, and (iv) if a category's kernel contains a specific region.

As an example let us consider the conceptual spaces described in Figure 3, below.

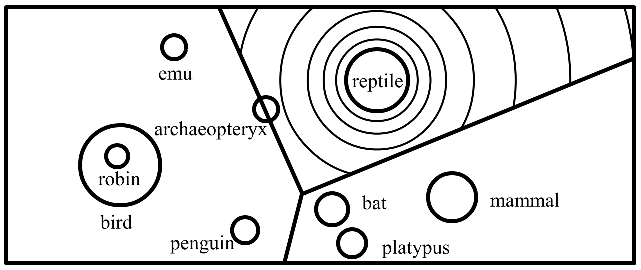

Figure 3: *Prototype regions in animal space, reptile crispings, & the power categorization of bird, mammal &reptile.*

Using RCC5 we can describe the following spatial relationships:

$DR(bird, penguin)$, $PP(robin, bird)$,
$PO(c(robin), bird)$, $PO(c(robin), kernel(c(bird)))$,
$PO(archaeopteryx, c(reptile))$,
$PO(archaeopteryx, c(bird))$,
$DR(c(archaeopteryx), c(mammal))$,
$DR(emu, penguin)$,
$DR(c(archaeopteryx), c(mammal))$,
$DR(c(robin), c(bat))$, and $DR(c(robin), c(platypus))$

The RCC, including the egg-yolk theory, allows disconnected regions, i.e. multi-pieced regions, so it supports the construction of arbitrarily complex concepts. For conceptual spaces that means one can juxtapose disconnected concepts to form eclectic ones, e.g. penguins and emus (nonflying birds), and build new concepts from existing ones. Within a single category the set of prototypes would in some applications be better modeled by a multi-pieced region rather than as a connected region such as the smallest surrounding sphere, or convex hull. In Section 5.3 we show that being able to model multi-pieced regions is important to support nonmonotonic reasoning.

For some applications it will be necessary to impose various ontological constraints on interrelated categories. In particular, it may be important to enforce consistency across the different levels of granularity so that the tessellated regions at one level are identical to the union of tessellated regions at lower levels: $c(\mathbf{p}) = \bigcup_{s \subseteq c(\mathbf{p})} c(s)$ where s are subcategories of $c(\mathbf{p})$. This constraint is present in many software engineering applications, and made explicit in data modeling techiques. One way to model this constraint in a computationally efficient manner (without distorting the underlying similarity relation) is to bound tessellations within categories. It is important to note here that $\Delta(\mathbf{P}, d, \mathbf{C}_{\cap S})$ is not identical to $\Delta(\mathbf{P}, d, S)$ in general, so bounding a conceptual space before or after the tessellation can, and typically will, result in a different categorization. For example, if the conceptual space parameters remain fixed then it would seem reasonable that the bird category region be the same regardless of whether the generator is the prototypical bird or the set of all prototypical birds at a lower level of granularity. In Figure 3 $c(bird)$ can be tessellated independently of $c(reptile)$ and $c(mammal)$, so that $c(bird) = c(robin) \cup c(penguin) \cup c(emu) \cup (c_{\cap c(bird)}(archaeopteryx))$. In other words, the generated subcategories of $c(bird)$ are bounded by $c(bird)$.

Other applications may possess weaker ontological requirements such as: If the prototype region P_1 is a subregion of the prototype region P_2 then $c(P_1) \subseteq c(P_2)$. This condition also places constraints on the way that a Voronoi tessellation can be generated across the levels of granularity, and is satisfied by $robins$ and $birds$ in Figure 3.

5.2 Crisping Conceptual Spaces

As noted in section 3 the RCC was extended by introducing an irreflexive, asymmetric and transitive binary relation $X < Y$ read as "X is crisper than Y", or "Y is a blurring of X". Cohn and Gotts also developed what has become known as the "Egg-Yolk Theory"for modeling indeterminate spatial regions. An *egg* is composed of two regions with definite boundaries; the *yolk* being a proper part of the egg. The egg and its yolk define the upper and lower bounds, respectively, on the range of indeterminacy of the region.

In this section we show how the crisper relation $<$ can be defined and the egg-yolk representation can be used to reason about categories and other conceptual regions.

A crisper relation in conceptual space can be constructed in a multitude of ways. One straightforward method is to use the proper part relation, PP, i.e. $X < Y$ if and only if $PP(X, Y)$. Figure 3 also illustrates some potential crispings of the $reptile$ category; a set of concentric spheres bounded by $c(bird)$ and $c(mammal)$. These crispings could be generated in numerous ways, e.g. using prespecified degradations in the distance measure, or by using the exemplars where each successive blurring captures another exemplar.

Given a conceptual space and a crisper relation we can build a vast range of useful queries using the RCC such as "Does a conceptual region constitute a crisping/blurring of another region?" "Does crisping a particular domain change the classification of a specific object?", "Does every category contain its prototype crisping?" and so forth.

As an example let us consider the conceptual space in Figure 3 where the crisper relation is based on PP. We have the following:

$penguin < c(penguin) < kernel(c(bird)) = c(bird)$
$robin < bird$ and $robin < c(robin) < c(bird)$
$emu < c(emu) < c(bird)$
$bat < c(bat) < c(mammal)$
$platypus < c(platypus) < c(mammal)$

Other relations describing the relationships between indeterminate regions can be constructed from the crisper relation $<$ such as $crisp(X)$ which is defined as "there does not exist a Y such that $Y < X$", and $MA(X, Y)$ which holds when X and Y are mutually approximate, i.e. they possess a common crisping [Cohn *et al.*, 1997]. From the conceptual space in Figure 3 we can say:

$Crisp(robin)$, $Crisp(mammal)$,
$Crisp(archaeopteryx \cap C(bird))$, $Crisp(penguin)$,
$\neg Crisp(bird), \neg Crisp(kernel(bird)), \neg Crisp(c(reptile))$,
$MA(c(penguin), c(bird))$, $MA(robin, bird)$,
$MA(archaeopteryx, c(bird))$,
$MA(archaeopteryx, c(reptile))$,

The RCC framework provides a number of axioms that govern the crisping relation in different kinds of applications. For example, there are axioms that ensure the existence of a complete crisping of any region, and the existence of alternative crispings and blurrings.

Crispings can play a role in the process of categorization itself; they can define regions to be used to generate tessellations. For example, in Figure 4, below, the bisector shifts towards the sphere P_1 with center $\mathbf{p}_1$ and radius r if P_1 is crisped to a smaller sphere P_1' with radius r'. This crisping can be modeled precisely; the bisector between P_1 and P_i moves by distance $(r - r')/2d(\mathbf{p}_1, \mathbf{p}_i)$ towards $\mathbf{p}_1$ in parallel to its previous location. So it is easy to show that $P_1' < P_1$ if and only if $c(P_1') < c(P_1)$, i.e. a local crisping (blurring) of a prototype region crisps (blurs) its category, and conversely.

In well behaved categorizations one can construct an egg yolk system using the kernel of each category. The yolk of the prototype region P_1 can be given by the largest sphere enclosed by $kernel(c(P_1))$, say $protoyolk$, and the egg can be given by the smallest sphere circumscribing $kernel(c(P_1))$, say $protoegg$. This egg-yolk prototype system can then be used to generate the corresponding egg-yolk category system: $c(protoyolk)$ and $c(protoegg)$.

Finally we extend the notion of crisping to distance measures to capture categorical perception; the observed phenom-

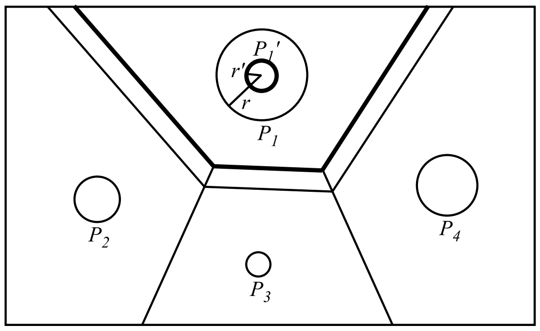

Figure 4: *Crisping a prototype crisps the category.*

ena where category members are judged to be more similar than members and nonmembers immediately across category boundaries. For example, the *distance measure d' is a crisping of distance measure d* where

$$d'(\mathbf{a},\mathbf{b}) = \begin{cases} k(d(\mathbf{a},\mathbf{b})) \text{ if } \mathbf{a},\mathbf{b} \in c(P) \text{ for some } P \\ d(\mathbf{a},\mathbf{b}) \text{ otherwise} \end{cases}$$

for some $0 \leq k < 1$.

In the limit we have $d'(\mathbf{a},\mathbf{b})$ is 0 if **a** and **b** are in the same category and $d(\mathbf{a},\mathbf{b})$ otherwise. Categorical perception, and hence crisping a distance measure, represents a form of learning. Our definition can be extended in numerous ways and the similarity relation derived from a crisped distance measure can easily be given a wide variety of threshold behaviours.

5.3 Nonmonotonicity and Concept Management

In this section we highlight the nonmonotonic effects of changing context, and show how conceptual spaces can be used as an underlying model from which more traditional nonmonotonic reasoning formalisms can be derived.

Nonmonotonic changes to the categorization can arise in several ways: (i) by *focusing* on a region, (ii) by *modifying* the underlying conceptual space, or (iii) by *changing* the mapping of objects to conceptual regions.

Focusing on a Region

Focusing can be accomplished by changing the dimension weights, by crisping a region, bounding a region, or combining regions.

As noted earlier categorization is context-sensitive. In conceptual spaces context-dependence is modeled using weighted dimensions. For example, weighting the distance measure along the x-axis results in a different categorization via the Voronoi tessellation such that objects change categories. Technically this is achieved by multiplying the specific dimensions by a given weight where the weight reflects its salience. For instance, under the Euclidean metric a weight can be placed on dimension i as follows: $\sqrt{\sum_i w_i(x_i - p_i)^2}$. Weighting specific dimensions gives rise to a nonmonotonic crisping relation where $X < Y$ does not imply $PP(X,Y)$; some regions will contract in size others will dilate. In Figure 5, below, the quality dimension represented by the x-axis in (a) becomes more salient and is elongated causing the object "**q**" to be reclassified in (b).

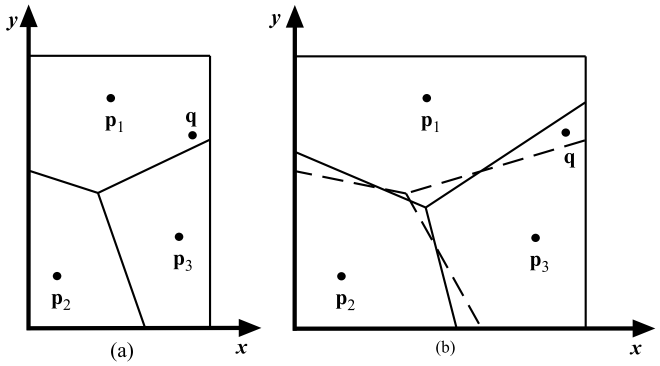

Figure 5: *Changed categorization due to a change in context.*

Modifying the underlying Conceptual Space

The underlying conceptual space can be modified by adding and deleting exemplars or prototypes, changing the distance measure, changing the function relating the distance measure to similarity, or changing the generator prototype regions.

The introduction of exemplars and/or prototypes will shift the boundaries and create new categorical regions [Gärdenfors, 2000]. As noted earlier the addition and removal of prototypes is a fundamental concept management operation, and can be achieved in linear time in the worst case.

Changing the underlying distance measure or generating regions will shift boundaries. Merely crisping the distance measure or modifying the function relating the distance measure to similarity will not change the underlying categorization, but will affect the magnitude of the similarity judgments.

Changing the mapping of objects to regions

Nonmonotonic reasoning formalisms are typically logic based, and hence symbolic systems. The conceptual space framework can be used to model nonmonotonic information, and used to construct nonmonotonic inference rules via the RCC. Since the conceptual space model is based on the measure of similarity to prototypes the RCC's crisper relation can be used to build representations of minimal models or usual states of affairs.

Just as specific individual objects are points in conceptual space, generic (or under specified) objects are (possibly disconnected) regions. One might expect that the more generic or the more unspecified an object, the larger the region used to represent it.

The generic bird, Tweety, would be represented as a central region in bird space, e.g. the prototype $bird$ region. In which case we would expect Tweety to possess all the features common to birds in that region; we expect Tweety to possess feathers, two legs, wings for flight, a four chambered heart, and so forth. As we learn more about Tweety we adjust the target region used to represent him. If we learn he is a robin, then he could be remapped to the $robin$ prototype. On the other hand, if we learn that he is a nonflying bird, then we may remap him to the prototype $penguin$ region and the prototype emu region which in the example in Figure 3 is a disconnected region, and we still expect Tweety to have feathers, two legs, and a four chambered heart.

The RCC can represent conceptual regions which are required to support all concept management for nonmonotonic reasoning as described above, and as such it forms a natural bridge from the geometrical conceptual space representation to the symbolic representation in standard nonmonotonic formalisms.

6 Conclusion

We showed how algorithms in computational geometry and the RCC can be applied to the conceptual space framework. Categorization in conceptual spaces is achieved via (generalized) Voronoi tessellations based on a similarity relation which results in a prototype being centrally located in its category. An analysis of existing algorithms in computational geometry established that categorization based on an underlying similarity relation which is used to generate Voronoi diagrams is feasible. Furthermore, once a categorization is completed concept management tasks like determining the category of an arbitrary object, adding and deleting prototypes, and merging categories are computationally fast. For example, once a classification has taken place the time taken to identify the category of an arbitrary object is logarithmic, and the time required to add or delete a prototype is linear.

We demonstrated that the ability to reason about categories and other conceptual regions using the RCC enhances the conceptual space model. We showed that the RCC can be used to construct both monotonic and nonmonotonic reasoning systems from the information embedded in the categories of conceptual spaces, hence the RCC forms a natural bridge from the geometrical representation of information in conceptual spaces to its symbolic counterpart.

In addition, the RCC provides facilities to determine the prototype regions from which to generate categories so that well behaved categorizations are produced.

The crisper relation $<$ and egg-yolk systems can be generated in conceptual spaces using prototypes and exemplars, and hence can be used to model the indeterminacy of conceptual regions. Interesting properties followed from our constructions, for example, we were able to show that crisping a prototype region locally, leads to the crisping of its category under the power categorization, and that categorical perception can be explained by crisping the distance measure on the conceptual space.

References

[Aurenhammer, 1987] Aurenhammer, F., Power Diagrams: Properties, Algorithms and Applications, *SIAM Journal of Computing Surveys*, 16(1):78-96, 1987.

[Aurenhammer, 1991] Aurenhammer, F., Voronoi Diagrams: A Survey of a Fundamental Data Structure, *ACM Computing Surveys*, 23(3), 345 - 405, 1991.

[Balkenius, 1999] Balkenius, C., Are There Dimensions in the Brain? in *Spinning Ideas, Electronic Essays Dedicated to Peter Gärdenfors on His Fiftieth Birthday* online at http://www.lucs.lu.se/spinning.

[Borsuk and Szmielew, 1960] Borsuk, K., and Szmielew, W, *Foundations of Geometry*. Amsterdam: North Holland, 1960.

[Bennett, 1997] Bennett, B., Logical Representations for Automated Reasoning about Spatial Relationships, *PhD Thesis*, University of Leeds, 1997.

[Chella *et al.*, 1998] Chella, A., Frixione, M., and Gaglio, S. An architecture for autonomous agents exploiting conceptual representations, *Robotics and Autonomous Systems*, 25(3-4):231–240, 1998.

[Cohn *et al.*, 1997] Cohn, A. G., Bennett, B., Gooday, J. and Gotts. N.M., Qualitative Spatial Representation and Reasoning with the Region Connection Calculus, *Geoinformatica*, 1(3), 1-42, 1997.

[Cohn and Gotts, 1996] Cohn, A. G. and Gotts. N.M., Representing Spatial Vagueness: A Mereological Approach, in *Principles of Knowledge Representation and Reasoning: Proceedings of the 5th International Conference*, Morgan Kaufmann, San Francisco, 230-241, 1996.

[Dwyer, 1987] Dwyer, R.A., Higher-Dimensional Voronoi Diagrams in Linear Expected Time, in *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*, 326 - 333, 1987.

[Gärdenfors, 2000] Gärdenfors, P., *Conceptual Spaces: The Geometry of Thought*, A Bradford Book, MIT Press, Cambridge Massachusetts, 2000.

[Harnad, 1987] Harnad, S., *Categorical Perception: The Groundwork of Cognition*, Cambridge University Press.

[Hampton, 1993] Hampton, J., Prototype Models of Concept Representation, *Categories and Concepts: Theoretical Views and Inductive Data Analysis*, Academic Press, London, 67 - 95, 1993.

[Krusal, 1978] Krusal, J.B., and Wish, M, *Multi-dimensional Scaling*. Beverley Hills, CA: Sage Publications, 1978.

[Klee, 1980] Klee, V., On the complexity of d-dimensional Voronoi diagrams, *Archiv de Mathematik*, 34: 74 - 80, 1980.

[May *et al.*, 1989] May, B. Moody, D.B., Stebbins, W.C., Categorical perception of conspecific communication sounds by Japanese macaques, Macaca fuscata , *J. Acoust. Soc. Am.*, Vol. 85, No. 2, Pages 837 - 847, 1989.

[Nebel, 1995] Nebel, B., Computational Properties of Qualitative Spatial Reasoning, in Wachsmuth, Rollinger and Brauer (eds), *Proceedings of the 19th German AI Conference*, Vol 981 LNCS, 233 - 244, Springer-Verlag, 1995.

[Okabe et al., 2000] Okabe, A., Boots, B., Sugihara, K., and Chiu, S.N., *Spatial Tessellations*, 2nd Ed, Wiley, 2000.

[Petitot, 1989] Petitot, J., Morphodynamics and the Categorical Perception of Phonological Units, *Theoretical Linguistics*, 15: 25 -71, 1989.

[Renz and Nebel, 1999] Renz, J. and Nebel, B., On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus, *Artificial Intelligence*, 108 (1-2): 69 -123, 1999.

[Rosch, 1975] Rosch, E., Cognitive representations of semantic categories, *Journal of Experimental Psychology: General*, 104: 192 - 232, 1975.

Simulating the Formation of Color Categories

Tony Belpaeme
Vrije Universiteit Brussel
Artificial Intelligence Lab
Pleinlaan 2, 1050 Brussels, Belgium
tony@arti.vub.ac.be

Abstract

This paper investigates the formation of color categories and color naming in a population of agents. The agents perceive and categorize color stimuli, and try to communicate about these perceived stimuli. While doing so they adapt their internal representations to be more successful at conveying color meaning in future interactions. The agents have no access to global information or to the representations of other agents; they only exchange word forms. The factors driving the population coherence are the shared environment and the interactions. The experiments show how agents can form a coherent lexicon of color terms and —particularly— how a coherent color categorization emerges through these linguistic interactions. The results are interpreted in the light of theories describing and explaining universal tendencies in human color categorization and color naming. At the same time the experiments confirm the view that certain aspects of language act as a complex dynamic system, arising from self-organization and cultural interactions.

1 Introduction

Color has enthralled scientists for centuries. Many disciplines in science, among which physics, neurology, cognitive science, philosophy, psychology, linguistics and anthropology, have all contributed to a vast body of work on the aspects of human color vision, including color perception, color categorization and color naming. Cognitive processes concerned with color have often been considered as being ideal test ground for verifying theories proposed in the above-mentioned disciplines. Moreover, empirical studies on color perception have always offered ample food for thought for quite a few different opinions in cognitive science; often the interpretation being changed to better fit this or that perspective.

The experiments described here study color categorization and color naming in artificial, well-controlled simulations; trying to provide justification or even new insights in theories on color categorization and color naming.

1.1 From color perception to color categories

Human color perception can be studied at several levels. At the neurological level, electro-magnetic energy is transformed in the photoreceptors of the retina into a neural signal, which is then conveyed to the brain. Humans have three different types of color sensitive photoreceptors: one sensitive for reddish light, one for greenish and one for bluish light. The cells are cone shaped and they are respectively called the L, M and S-cones; designating their sensitivity to long, middle or short wavelengths. Humans are thus trichromatic species. However, at the psychological level humans seem to react rather different than one would expect for a species having three types of photoreceptors. Hering's opponent color theory, later defined quantitatively by [Jameson and Hurvich, 1955] and observed experimentally during in vivo experiments on macaque monkeys by [DeValois *et al.*, 1966], argued for an antagonistic nature of color perception: color seems to occur in pairs, with black opposed to white, green opposed to red and blue opposed to yellow. This gave rise to the two stage color theory; in a first stage light is received by three types of photoreceptors, and in the second stage the outputs are interconnected to form red-green, yellow-blue and white-black channels. However, we are still left with a continuous color experience, handling color information would require cutting up that color continuum. This brings us to color categorization.

1.2 Color categories and color terms

Color appearance has a categorical nature; this is immediately suggested by the fact that every language has different color words to indicate different color sensations. The belief was long held that cultures divided the color spectrum into arbitrary categories. However, in 1969 Berlin and Kay published their influential monograph [1969], in which they provide empirical evidence for universal tendencies in color categorization. They conclude that humans have eleven basic perceptual color categories; *basic* meaning that the corresponding color term is a monolexemic, unique color term, salient and unambiguous to all language speakers. Human languages have at least two and at most eleven basic color terms referring to these perceptual color categories (English has all eleven of them: black, white, red, green, yellow, blue, brown, purple, pink, orange and gray). A second conclusion is that basic color terms appear in languages in a specific order. When a language has only two color terms it will be a term for BLACK and WHITE, when a third color term is added, it will be RED, next either GREEN or YELLOW is lexicalized and so on. At about the same time, new quantitative information on the opponent character of color vision seemed to support Berlin and Kay's theories very well [Kay and McDaniel, 1978]. Thus, the universalist stance quickly became widely accepted. However, recently critical views have been offered on universalist extremism, pleading for a more subtle attitude and for more carefully collected and interpreted quantitative data [Saunders and van Brakel, 1997; Lucy, 1997].

1.3 The relation to language

Language is unique to humans; although many animals are capable of communicating messages, they are not able of employing the full range of linguistic capabilities as we humans can. Concrete and abstract concept formation, extensive lexicalizations and syntax all seem to be exclusive to humans. The way humans handle abstract reasoning, hierarchical structures and arbitrary mapping is unsurpassed, and there is good reason to believe that language is crucial to all this. The nature of language and the origin of language might indeed help us understand human intelligence.

On the origin of language, two extreme stances exist. Some assume that human language capacity is innate and at large genetically defined [Chomsky, 1980; Pinker and Bloom, 1990; Bickerton, 1998], while others believe that the origins of language are to be found in the combined play of human general cognitive capacities and cultural interactions [Deacon, 1997; Steels, 1999].

Steels [1997; 1998] considers language to be the product of cultural evolution. According to Steels language can be seen as a distributed, dynamical and adaptive system. Language is not controlled by one central intelligence; instead, the knowledge of the language is distributed over its users. None of the users has perfect knowledge or control of the language. Language is also robust to changes in the population; users may leave or join the community without significantly affecting the language spoken in the community. In addition language can be seen as a complex dynamic system: categories, concepts, word forms and grammar emerge and adapt according to population dynamics which can be described using ideas from the field of dynamical systems. These theories have been successfully used, for example to explain the self-organization of universal tendencies in vowel systems [de Boer, 2001].

Investigating the formation of color categories and color terms in the same way can help elucidate some aspects of human language, such as concept formation, lexicon grounding and the propagation of lexicalizations and meaning through a population.

The paper is structured as follows. Section 2 describes the internal organization of the individual agents (the representation of color perception, the categorization and the connection between color word and color categories). Section 3 provides details on the dynamics on the individual level and on the population level. Section 4 provides results illustrating some typical outcomes of simulations, while section 5 and 6 conclude.

2 The agents

The simulation uses a population of agents. On an individual level, the agents all have the ability to perceive color, to categorize their perception, to lexicalize their color categories and to adapt to other agents in order to be more successful at communicating color meaning. On the population level, the agents communicate with each other using a simple protocol called the guessing game. In a guessing game, two agents communicate about a visual context. Through playing several thousands of these games a common lexicon is built up.

2.1 Color perception and representation

When perceiving the physical world, a mapping is made from the physical space to a representation in the psychophysical space. Upon this representation, further cognitive actions such as categorization or recognition are taken.

The color stimuli are presented to the agents as spectral power distributions, expressed as energy at wavelengths in the visible spectrum (ranging from 380 nm to 800 nm). No information on spatial or temporal properties are given: the colors are presented in "aperture" mode, void of any contextual information. The sensation $S \in \mathbf{S}$ is a physical stimulus and has to be mapped to a psychophysical representation $R \in \mathbf{R}$. For this a mapping $\mathbf{S} \rightarrow \mathbf{R}$ is needed. The representation $\mathbf{R}$ should fulfill three requirements. First and for all it should be a good model for how humans perceive color. Second, it should make discrimination possible: two stimuli are discriminable if and only if they map onto different points in the representation space. Third, one should be able to define a similarity measure over the representation space.

The CIE $L^*a^*b^*$ color space satisfies these requirements and has proven its merit at categorizing real-world color images (see [Lammens, 1994]). It is a three-dimensional color space designed to be perceptually equidistant and can be represented in Cartesian space. L^* represents lightness, a^* corresponds approximately to redness-greenness and b^* to yellowness-blueness. For a definition and for a conversion from spectral power distribution to CIE $L^*a^*b^*$ see for example [Wyszecki and Stiles, 2000])

2.2 Color categorization

When an agent is to communicate about the world, a symbolic representation of the perception is needed. This categorical representation arises by cutting up and structuring the representation space.

The color space (which is the psychophysical representation space in these experiments) is used to define categories. A category has a number of features, in our case L^*, a^* and b^*, and for each feature a fuzzy membership function is defined. If an unknown stimulus is perceived, a measure is needed of how well a category matches the unknown representation. Several solutions are possible to represent a category, one could take a radially symmetric function such as a Gaussian; or a multilayer perceptron could be used. Here an adaptive network is chosen (an adaptive network resembles a radial basis function —see eg. [Broomhead and Lowe, 1988]; except that there is only one output unit, which is divided by the number of hidden units). Adaptive networks are our preferred choice for representing categories since they can divide the input space into regions while not being restricted in any way: the regions can be convex or not, symmetrical or not, connected or disparate, even overlap is possible. A second advantage is that adaptive networks are easily analyzed. This is valuable for monitoring the performance of the simulations.

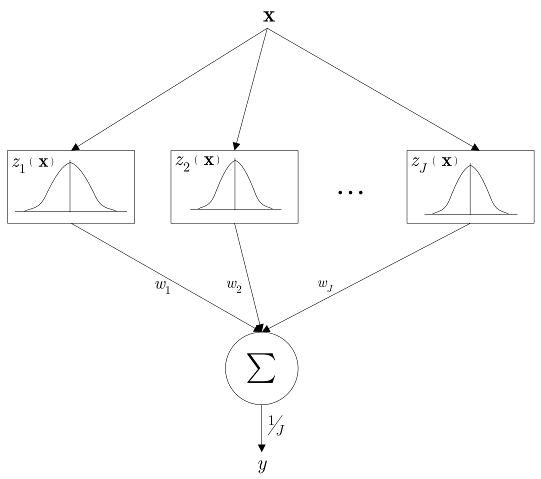

Figure 1: adaptive network for representing a color category, it consists of one hidden layer of locally tuned receptors fully connected to a linear output unit.

Figure 1 shows the adaptive network. It consists of a layer of an unspecified number of hidden units acting as tuned receptors and one output unit. The input $\mathbf{x}$ is a three-dimensional vector containing a L^*, a^* and b^*-value. The hidden units are Gaussian functions $z_j(\mathbf{x})$, with center $\mathbf{m}_j$ and width σ_j. The output of the network $y(\mathbf{x})$ is the weighted sum of the Gaussians, weighted by the number of hidden units.

$$z_j(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x}-\mathbf{m}_j)^2}{2\sigma_j^2}\right)$$

$$y(\mathbf{x}) = \frac{1}{J}\sum_{j=1}^{J} w_j z_j(\mathbf{x})$$

For adapting the network, a combination of instance-based and reinforcement learning is used. There are four possible actions: adding or removing a hidden unit, and increasing or decreasing the weight w_j of a unit. The width σ_j of a unit —initialized to a default value— is never changed.

2.3 Lexicalizing categories

Finally, the agents need word forms in order to communicate about color categories: word forms are the only information exchanged by the agents. A category can be associated with one or more word forms, allowing for synonymy. It is also allowed for the same word form to be associated with more than one category, allowing for polysemy. Note that categories are only lexicalized when they need to be communicated; often agents have categories with no word form associated.

An agent has a set of associations A, each association is a pair containing a category $c_i \in C$ and a set of word forms F_i (F_i can be empty).

$$A = \{\langle c_1, F_1\rangle, ...\}$$

Word forms $f \in F$ are randomly selected from a finite alphabet V, $f \in V^*$. No other restrictions are applied to the creation of word forms (for example, it is not the case that more often used words tend to be shorter, as observed in human languages).

3 The simulation

During a simulation step two kinds of games are played. The *discrimination game* is played at the individual level. The *guessing game* is played at the population level. More on the mechanism of both games can be found in [Steels, 1998].

3.1 The discrimination game

The goal of the discrimination game is to construct categories in order to successfully distinguish color stimuli. It follows a simple scenario, and is completed by one agent without the need for interactions with other agents.

An agent has a, possible empty, set of categories C. A random context $O = \{o_1, ..., o_N\}$ is created and presented to the agent. It contains N objects o_i (in this case color stimuli) of which one object o_t is the topic. The topic has to be discriminated from the rest of the context. The game proceeds as follows.

1. Context $O = \{o_1, ..., o_N\}$ and the topic $o_t \in O$ are presented to the agent.
2. The agent perceives each object o_i and produces a sensory representation for each object: $S_{o_i} = \{s_1^{o_i}, ..., s_M^{o_i}\}$.
3. For all N sensory representations, the best matching category $c_{S_o} \in C$ is found.

   $$\forall c \in C : y_c(S_o) \leq \hat{y}(S_o)$$

   y_c is the output of the adaptive network belonging to category c, and $\hat{y}$ is the output of the adaptive network reacting best to S_o.
4. The topic o_t can be discriminated from the context when there exists a category matching the topic but not matching any other objects in the context.

   $$\left\{c_{S_{o_1}}, ..., c_{S_{o_N}}\right\} \cap c_{S_{o_t}} = \left\{c_{S_{o_t}}\right\}$$

This scenario can fail in two ways. First, the agent has no categories yet ($C = \emptyset$); in this case a category is created with its center on the topic. Second, no discriminating category can be found: the category found for the topic is also found for other objects. When this category is far from the topic (according to some distance measure), a new category is created. When it is closer than a certain threshold distance, the category is adapted by adding a new hidden unit with its center on the topic.

When playing these discrimination games an agent is able to create categories that discriminate one object (i.e. color stimuli) from others. Next to basic mechanism, the weight of the hidden units are decreased with every game. Over time, this results in the "forgetting" of hidden units. Only when a category has proven to be useful in an interaction, the weights are increased again.

3.2 The guessing game

For the guessing game, two agents are randomly chosen. One acts as the *speaker*, the other as the *hearer*. A context O is presented to both agents, but only the speaker knows the topic. The game goes along the following scenario.

1. The speaker tries to discriminate the topic by playing a discrimination game, if it finds a discriminating category $c_{S_{o_t}}$ the game continues, otherwise the game fails.
2. The speaker looks if any word forms are yet associated with $c_{S_{o_t}}$. If not, a new word form f is randomly created and associated. If however one or more word forms are already associated with $c_{S_{o_t}}$, then one word form f is selected according to its success in previous guessing games. The word form f is then conveyed to the hearer.
3. The hearer looks if it has f in its associative memory, if not the game fails: the hearer is shown the topic o_t and it learns the proper word form for it by adding a category for the topic.
4. If the hearer does have the word form f in its lexicon it finds the associated category c' and tries to point out the topic. This will only work when the hearer can discriminate the topic from the context, otherwise the game fails.
5. If the hearer succeeds in pointing out the correct object as the topic, the game is successful. If the hearer points out the wrong object, the speaker identifies the topic and the hearer adapts its category c' by adding a hidden unit, so the category resembles the topic better in future games.

When a guessing game is successful the weights of the hidden units of the category $c_{S_{o_t}}$ are increased, this strengthens a category making the probability of it being used in future games higher. Categories which do not contribute to the game are through this less likely to be used in future interactions.

The interactions are responsible for achieving agent lexicons which are coherent over the population. And because the agents adapt their categories according to the outcome of the guessing game, the categories of the agents show coherence as well. This is an illustration of the Sapir-Whorf thesis, which claims that language influences the way its users experience the world [Whorf, 1950].

4 Results

For the simulations two data sets are used as input: one containing spectral measurements taken from over 1200 color chips of the Munsell color notation system and one with 300 measurements of colors of plants and flowers. A context is selected out of these databases, containing minimum 2 and up to 10 different color samples. As a reference, artificial created color samples are use to test the robustness of the system.

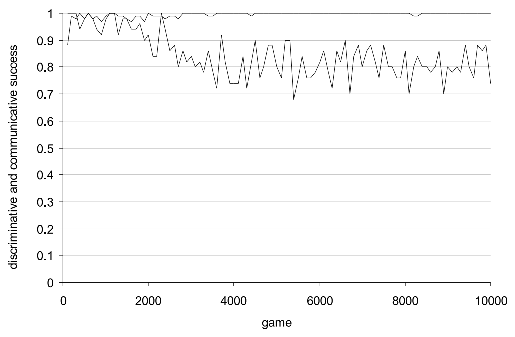

Figure 2: the average success rate plotted for 20 agents over 10000 games. The upper curve shows the discriminative success, the lower curve the communicative success.

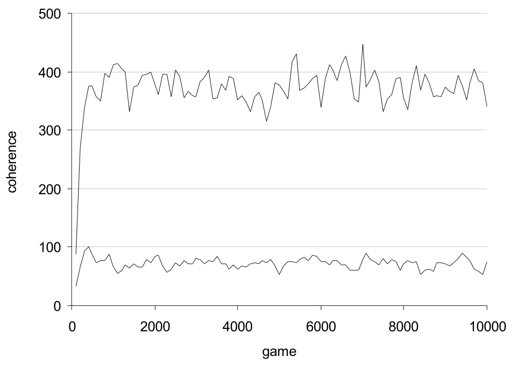

Figure 3: coherence of the color categories in a population. The bottom data series shows the coherence without interactions, the top series shows how interactions increase the coherence.

The results show that the agents create a set of categories with which they can discriminate any color context offered (provided that the color stimuli in the context are dissimilar enough; for example, the colors can not be metameric). Figure 2 shows the success rate of a population of agents during a typical run[1]. The discriminative success —telling how good the agents are at discriminating the context— quickly rises to 100%. The communicative success —measuring how good the agents are at conveying meaning— rises quickly, then drops off as the games reach their full complexity and then gradually rises again as they agree on a common lexicon.

Another result demonstrates the benefit of communication. When no guessing games are played, so that there is no interaction between the agents, the agents do not manage to

[1]Note that similar phenomena are observed for a wide variety of parameter settings, the limited space however does not allow extensive reporting of results.

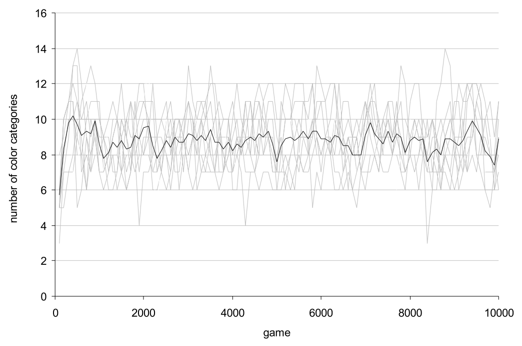

Figure 4: number of categories for 10 agents during 10000 games. The gray curves show the number of categories for each agent, the black curve shows the population average.

form coherent category sets. Clearly the environmental binding is not strong enough to obtain shared categorizations. When the interactive component is introduced, by letting the agents play guessing games, the coherence of the color categories rapidly increases. The coherence is computed by cross-summing the similarity for all categories of the entire population. Let $C' = \cup C$ contain all the categories of all the agents, the coherence is then computed as,

$$\text{coh} = \sum_{i=1}^{C'} \sum_{j=i+1}^{C'} \text{sim}\left(c'_i, c'_j\right)$$

The higher the coherence, the better the categories of the agents agree. Categories are matched according to a similarity function. The similarity between two categories c_a and c_b is computed as in eq. 1, with J_{c_a} and J_{c_b} being the number of hidden units for both categories. It is proportional to the inverse of the Euclidean distances between the centers of the hidden units of both categories.

$$\text{sim}(c_a, c_b) = \frac{1}{J_{c_a} J_{c_b} \sum_{i=1}^{J_{c_a}} \sum_{j=i+1}^{J_{c_b}} \left\| \mathbf{m}_{c_a,i} - \mathbf{m}_{c_b,j} \right\|} \quad (1)$$

Figure 3 shows a typical run, with a population of 20 agents playing 10000 games, the context contains 5 color stimuli selected from the Munsell database. The coherence of a population only playing discrimination games is low compared to a population benefiting from interactions through guessing games; it demonstrates how linguistic interactions are responsible for coherence of categories. This might seem surprising because the agents never have access to the categories of other agents. However, through the linguistic interactions their internal representations adapt to allow for improved transfer of meaning. A side effect of this is that the categories become coherent over the population.

The number of color categories (and proportionally the number of associated word forms) created by the agents rises quickly and stabilizes after a while. Figure 4 shows a typical run: the number of categories stabilizes on an average of 9.4 color categories per agent. The number of color categories depends on several parameter settings. Parameters having a large influence on the number of color categories are (1) the number of color samples in the context, more color samples force the agents to create more color categories to be able to discriminate the colors and (2) the similarity of the color samples; if the samples are rather similar, fine grained categories are needed to discriminate them. In a nutshell, the context exerts pressure on the agents to create more or less categories[2].

The environment does not only influence the quantity of categories, it also affects their quality. When the context contains only highly saturated colors, the agents will only create categories presenting high saturated colors. Likewise, when the context contains a significantly higher amount of red samples, the agents are likely to all have a category and word form for red.

5 Discussion

The evolutionary order of the emergence of named color categories, as observed by Berlin and Kay, does not show up in the experiments where there is no bias imposed on the color perception or on the environment. When in simulation the agents have only two color terms, they will have a term for warm-bright colors and one for dark-cool colors, which is in accordance with observations of human languages. However, when the agents have three or more color terms, there is no preference for creating a category for reddish colors: the creation of categories is entirely opportunistic. For humans the story is different, when human languages have three or more color terms, there will always be a term for reddish colors. Several explanations have been offered for this (see [Hardin, 1987] for an overview) that can be summarized in two ideas: the preference could be built into human biology or it could be rooted in near-universal environmental constraints. For years the focus has been on the former: by interpreting the neurophysiological structure of human color perception one is able to explain many observed phenomena of color lexicalizations. However, some discrepancies remain which can not be explained by the neurological makeup of color perception. For instance, why do some languages not follow the evolutionary order proposed by Berlin and Kay? Why do some languages have more than one word for blue? And why do languages spoken around the equator have less color terms? See [Saunders and van Brakel, 1997].

Although a strong caveat should be issued when generalizing from artificial simulations to real-world phenomena, simulations can sometimes offer new insights and might help us find new ways to tackle problems. The simulations show how in simple artificial linguistic setting, a coherent color lexical and color categorizations can emerge in population of agents. This does not mean that shared human color categories emerge through cultural interactions; there is evidence that already infants have a categorical preference for certain

[2]This bears resemblance to the folk theories on why equatorial languages have less color terms; it seems language communities there have experienced less pressure to extend their color vocabulary because color technology has evolved more slowly and has only been fairly recently brought up to Western standards.

strong hues, long before they engage in linguistic interactions [Bornstein, 1973]. However, it might very likely that it is the interplay between cultural dynamics and biological dispositions that is responsible for how we categorize color, and not color alone.

6 Conclusion

The experiments show how out of self-organization and linguistic interactions a coherent color lexicon can emerge. In addition, it shows how color categories can become shared among a group of language users solely by linguistic interactions. The color lexicons and category sets stabilize under a large range of parameter settings, showing the robustness of the system. The agents and their interaction dynamics form a dynamic system, with attractors in the form of stable lexical and category sets.

Many things still need to be investigated. Most important, the influence of bias on the perception and on the environment needs to be studied. The conditions under which more realistic color categories arise, including the evolutionary order, need to be studied. Already it is clear that the environment and contextual information will play an important role in this. As a bonus, it might be interesting to see if the system could learn color categories and names from a human instructor.

Acknowledgments

I would like to thank Erik Myin and Jelle Zuidema for their useful comments. The author is a Research Assistant of the Fund for Scientific Research - Flanders (Belgium) (FWO).

References

[Berlin and Kay, 1969] Brent Berlin and Paul Kay. *Basic Color Terms. Their Universality and Evolution*. University of California Press, Berkeley, CA, 1969. Reprinted in 2000.

[Bickerton, 1998] Derek Bickerton. Catastrophic evolution: the case for a single step from protolanguage to full human language. In James R. Hurford, Michael Studdert-Kennedy, and Chris Knight, editors, *Approaches to the Evolution of Language*, pages 341–358. Cambridge University Press, 1998.

[Bornstein, 1973] Marc H. Bornstein. Color vision and color naming: a psychophysiological hypothesis of cultural difference. *Psychological Bulletin*, 80(4):257–285, 1973.

[Broomhead and Lowe, 1988] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[Chomsky, 1980] Noam Chomsky. Rules and representations. *Behavioral and Brain Sciences*, 3:1–21, 1980.

[de Boer, 2001] Bart de Boer. *The origins of vowel systems*. Oxford University Press, 2001. To appear.

[Deacon, 1997] Terrence W. Deacon. *The Symbolic Species: The Co-evolution of Language and the Brain*. Norton, 1997.

[DeValois *et al.*, 1966] Russell L. DeValois, I. Abramov, and G.H. Jacobs. Analysis of response patterns of lgn cells. *Journal of the Optical Society of America*, 56(7):966–977, 1966.

[Hardin, 1987] C.L. Hardin. *Color for philosophers*. Hacket Publishing Company, Indianapolis, 1987.

[Jameson and Hurvich, 1955] Dorothea Jameson and Leo M. Hurvich. Some quantitative aspects of an opponent-colors theory. i. chromatic responses and spectral saturation. *Journal of the Optical Society of America*, 45(7):546–552, 1955.

[Kay and McDaniel, 1978] Paul Kay and Chad K. McDaniel. The linguistic significance of the meaning of basic color terms. *Language*, 54:610–646, 1978.

[Lammens, 1994] Johan M. Lammens. *A computational model of color perception and color naming*. PhD thesis, State University of New York, Buffalo, 1994.

[Lucy, 1997] John A. Lucy. The linguistics of “color”. In Clyde L. Hardin and Luisa Maffi, editors, *Color Categories in Thought and Language*, pages 320–346. Cambridge University Press, 1997.

[Pinker and Bloom, 1990] Steven Pinker and Paul Bloom. Natural languages and natural selection. *Behavioral and Brain Sciences*, 13:707–784, 1990.

[Saunders and van Brakel, 1997] Barbara Saunders and Jaap van Brakel. Are there nontrivial constraints on colour categorization? *Behavioral and Brain Sciences*, 20:167–228, 1997.

[Steels, 1997] Luc Steels. The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–35, 1997.

[Steels, 1998] Luc Steels. Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. In James R. Hurford, Michael Studdert-Kennedy, and Chris Knight, editors, *Approaches to the Evolution of Language*, pages 384–404. Cambridge University Press, 1998.

[Steels, 1999] Luc Steels. The puzzle of language evolution. *Kognitionswissenschaft*, 8(4), 1999.

[Whorf, 1950] B.L. Whorf. *Four articles on metalinguistics*. Foreign Service Institute, Washington, 1950.

[Wyszecki and Stiles, 2000] Gunther Wyszecki and W.S. Stiles. *Color science : concepts and methods, quantitative data and formulae*. John Wiley and Sons, 2nd edition, 2000.

COGNITIVE MODELING

COGNITIVE MODELING — PERCEPTUAL GROUNDING

Grounded Models as a Basis for Intuitive Reasoning

Josefina Sierra-Santibáñez
Escuela Técnica Superior de Informática
Universidad Autónoma de Madrid
josefina.sierra@ii.uam.es

Abstract

This paper introduces *grounded models* and compares them to axiomatic models of mathematics. Grounded models differ from axiomatic theories in establishing explicit connections between language and reality that are learnt through language games. They are constructed and updated by autonomous agents connected to their environment through sensors and actuators using some conceptualization mechanisms and language games described in [Steels, 1999]. They are based on conceptualization and support a form of *intuitive reasoning*, which can be done sometimes by constraint satisfaction and it is argued to be the basis of some axiomatizations. This is illustrated with a simple example of spatial reasoning.

1 Introduction

The concept of *grounded model* introduced in this paper is based on some ideas and mechanisms described in *The Talking Heads Experiment* [Steels, 1999]. The experiment involves a set of robotic ”Talking Heads” playing *language games* with each other about scenes perceived through their cameras on a white board in front of them. In particular, we focus on the conceptualization part of a language game, called the discrimination game, which generates the meaning of the verbal hint transmitted by the speaker to the hearer in a language game.

The *discrimination game* [Steels, 1996] is played by a single agent, and consists of the following steps. First, the agent perceives an image on a white board through his camera, segments the image into coherent units, and computes various sensory characteristics about each image segment, such as its color, horizontal or vertical position. Then, the agent chooses an image segment as topic, and tries to find a combination of categories that distinguishes the topic from the other objects in the image. The game succeeds if the agent finds a combination of categories that is true for the topic, but it isn't true for the other objects in the image. If the game fails, the agent adapts its internal structures to become more successful in future games.

The rest of this section describes, in some detail, the different processes and cognitive structures involved in the discrimination game.

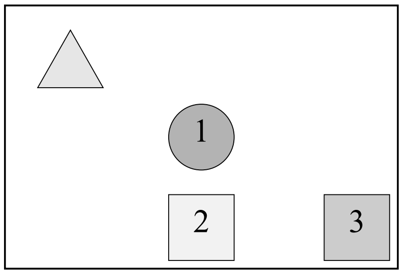

Figure 1: Each object in the scene is characterized by values on three primitive sensory channels: HPOS, VPOS and GREY.

1.1 Perception

The agent captures an image on a white board in front of him. Geometric figures of various sizes, shapes and colors can be pasted on the white board. Then, the agent segments the image into coherent units. Next low level visual processes gather information about each segment, such as its color, horizontal or vertical position. Each process outputs its information on a *sensory channel*. We assume that there are only three primitive sensory channels.

- HPOS(obj) contains the x-midposition of object obj.
- VPOS(obj) contains the y-midposition of object obj.
- GREY(obj) contains the average gray-scale of obj.

The values on the sensory channels HPOS, VPOS and GREY are scaled so that its range is the interval (0.0 1.0). Consider the three objects numbered in figure 1, object 1 has the values HPOS(Obj1)=0.5, VPOS(Obj1)=0.5, GREY(Obj1)=0.5, object 2 the values HPOS(Obj2)=0.5, VPOS(Obj2)=0.2, GREY(Obj2)=0.1, and object 3 the values HPOS(Obj3)=0.8, VPOS(Obj3)=0.2, GREY(Obj3)=0.3.

In addition to three primitive sensory channels, there are some sensory channels constructed from them.

- HPOS-DIFF(obj1,obj2) contains the difference of the x-midpositions of objects obj1 and obj2, i.e. HPOS-DIFF(obj1,obj2)=HPOS(obj1) – HPOS(obj2).
- VPOS-DIFF(obj1,obj2) contains the difference of the y-midpositions of segmented objects obj1 and obj2.

- GREY-DIFF(obj1,obj2) contains the difference of the average gray-scale of objects obj1 and obj2.
- EQUAL(obj1,obj2) is defined as a predicate, i.e. a function which takes the value 1 (i.e. true) if the values of the primitive sensory channels HPOS and VPOS are equal for obj1 and obj2, and 0 (i.e. false) otherwise.

For example, the sensory channel VPOS-DIFF has the value 0.3 when it is applied to the pair of objects formed by object 1 and object 2, i.e. VPOS-DIFF(Obj1,Obj2)=0.3. The range of the sensory channels HPOS-DIFF, VPOS-DIFF and GREY-DIFF is (-1.0 1.0). The range of the sensory channel EQUAL is the discrete set of Boolean values $\{0, 1\}$.

1.2 Perceptually Grounded Categories

The data on sensory channels are values from a continuous domain (except for sensory channel EQUAL). To be the basis of natural language communication, these values must be transformed into a discrete domain. One means of categorization is to divide up each domain of values on a particular sensory channel into regions and assign a category to each region. For example, the range of the HPOS channel can be cut in two halves leading to a distinction between [LEFT] ($0.0 <$ HPOS < 0.5) and [RIGHT] ($0.5 <$ HPOS < 1.0). Object 3 in figure 1 has the value HPOS=0.8 and would therefore be characterized as [RIGHT]. Similarly, the VPOS-DIFF channel can be cut in two halves as well leading to a distinction between [ABOVE] ($0.0 <$ VPOS-DIFF < 1.0) and [BELOW] ($-1.0 <$ VPOS-DIFF < 0.0).

It is always possible to refine a distinction by dividing its region. Thus an agent could divide the bottom region of the HPOS channel (categorized as [LEFT]) in two subregions [TOTALLY-LEFT] ($0.0 <$ HPOS < 0.25), and [MID-LEFT] ($0.25 <$ HPOS < 0.5). The categorization networks resulting from these consecutive binary divisions form *discrimination trees*.

We label categories using the sensory channel from which they operate, followed by the upper and lower bound of the region they carve out. Thus [TOTALLY-LEFT] is labeled as [HPOS 0.0 0.25], because it is true for a region between 0.0 and 0.25 on the HPOS channel. We assume that perceptually grounded categories correspond to n-ary predicates of first order logic. For example, the category [HPOS 0.0 0.25] corresponds to the unary predicate [HPOS 0.0 0.25](obj), and the categories [VPOS-DIFF 0.0 1.0] and [EQUAL] to the binary predicates [VPOS-DIFF 0.0 1.0](obj1,obj2) and [EQUAL](obj1,obj2).

There are other ways to move from the continuous domain of sensory channels to the discrete domain of categories. We could introduce focal values and associate a category with each of them. In this case, the categorization process consists in identifying the focal point that is closest to an object's value.

1.3 Concepts

Perceptually grounded categories can be combined to construct *concepts*. We use a set of concepts which can be defined by induction as follows.

1. If p is an n-ary category and $x_1, \ldots, x_n$ are variables, then $p(x_1, \ldots, x_n)$ is a concept.
2. If c is a concept, then the negation of c (written $\neg c$) is a concept.
3. If c_1 and c_2 are two concepts, then the disjunction of c_1 and c_2 (written $c_1 \vee c_2$) is a concept.

The symbols $\wedge$, $\rightarrow$ and $\leftrightarrow$ are introduced as abbreviations: (1) $c_1 \wedge c_2$ is an abbreviation of $\neg(\neg c_1 \vee \neg c_2)$; (2) $c_1 \rightarrow c_2$ is an abbreviation of $\neg c_1 \vee c_2$; (3) $c_1 \leftrightarrow c_2$ is an abbreviation of $(c_1 \rightarrow c_2) \wedge (c_2 \rightarrow c_1)$. Notice that the set of possible concepts is the set of free-quantifier formulas that can be constructed from the predicates associated with perceptually grounded categories.

We distinguish three types of concepts: intuitive truths, intuitive falsehoods, and regular concepts. Some concepts have the property of being true for every possible tuple of segmented objects. We will call them *intuitive truths* or just *truths*. For example, the concept [VPOS-DIFF 0.0 $1.0](obj1, obj2) \rightarrow$ $\neg$[VPOS-DIFF 0.0 $1.0](obj2, obj1)$ is an intuitive truth.

Other concepts have the property of being false for every possible tuple of segmented objects. We will call them *intuitive falsehoods*. For example, the concept [VPOS-DIFF 0.0 $1.0](obj1, obj1)$ is an intuitive falsehood.

The rest of the concepts, called *regular concepts* or simply *concepts*, can be used to discriminate those tuples of objects that satisfy them from those tuples of objects which do not make them true. For example, the concept $\neg$[VPOS-DIFF 0.0 $1.0](obj1, obj2) \wedge$ $\neg$[VPOS-DIFF 0.0 $1.0](obj2, obj1)$, which is true for a pair of segmented objects obj1 and obj2 if neither obj1 is above obj2, nor obj2 is above obj1, is a regular concept.

2 Grounded Models

2.1 Categorizers

A *categorizer* is a cognitive procedure capable of determining whether a category applies or not. For example, the behavior of the categorizer for category [VPOS 0.0 0.5](obj) can be described by a function that takes the value 1 if $0.0 <$ VPOS(obj) < 0.5, and 0 otherwise.

The categorizers of nonatomic concepts are compositions of the categorizers of categories. For example, the behavior of the categorizer for concept [VPOS-DIFF 0.0 $1.0](obj1, obj2)\} \wedge$
$\neg$[VPOS-DIFF 0.0 $1.0](obj2, obj1)$ can be described by a function that takes the value 1 if $0.0 <$ [VPOS-DIFF 0.0 $1.0](obj1, obj2) < 1.0$, and [VPOS-DIFF 0.0 $1.0](obj2, obj1) \leq 0.0$.

Categorizers establish explicit connections between symbols (categories and concepts) and reality (perceptual input as processed by sensory channels). These connections are learnt through language games [Wittgenstein, 1953], and allow agents to check whether a concept is true or not for a given tuple of segmented objects. Most importantly, they provide information on the perceptual and cognitive processes an agent must go through in order to understand or evaluate a given concept.

2.2 Grounded Models

A *grounded model* is the set of concepts and categorizers constructed by an agent at a given time in its development history. These concepts and categorizers are organized as follows.

1. A discrimination tree for every sensory channel containing categories and categorizers which use the values on that sensory channel.
2. A set of intuitive truths containing concepts which are true for every possible tuple of segmented objects.
3. A set of defined concepts containing nonatomic concepts which are neither intuitive truths nor intuitive falsehoods.

A grounded model reflects the conceptualization of the world constructed by an agent at a given time in its development history. Grounded models are not static, but evolve and adapt as the agent is confronted with new experiences.

Consider the grounded model G of an agent that has constructed top-level categories and categorizers for each sensory channel, but does not have any defined concept or intuitive truth yet. The categorizers of this grounded model are cognitive procedures whose behavior can be described by linear constraints. We associate a linear constraint describing the behavior of each categorizer with a predicate symbol that corresponds to the category it is capable of recognizing.

$$[\text{VPOS } 0.0\ 0.5](x) \equiv 0.0 < \text{VPOS}(x) \wedge \text{VPOS}(x) < 0.5$$

$$[\text{VPOS } 0.5\ 1.0](x) \equiv 0.5 < \text{VPOS}(x) \wedge \text{VPOS}(x) < 1.0$$

$$\vdots$$

$$[\text{GREY-DIFF } \text{-}1.0\ 0.0](x, y) \equiv$$
$$\text{-}1.0 < \text{GREY}(x)\text{-GREY}(y) \wedge \text{GREY}(x)\text{-GREY}(y) < 0.0$$

$$[\text{GREY-DIFF } 0.0\ 1.0](x, y) \equiv$$
$$0.0 < \text{GREY}(x)\text{-GREY}(y) \wedge \text{GREY}(x)\text{-GREY}(y) < 1.0$$

Categorizers are probably implemented by neural networks in natural agents. We only use linear constraints to model their behavior. We are not assuming therefore that agents do learn such constraints, but that they build them in their perceptual systems.

2.3 Grounded Models and Axiomatic Theories

A grounded model looks like an axiomatization or a theory of first order logic, but differs from it in some aspects. Like a logical theory, it has a set of basic concepts (which correspond to perceptually grounded categories), and a syntax that allows constructing defined concepts from basic concepts[1].

One of the most important differences between grounded models and logical theories is the form in which basic concepts are defined. In a grounded model, concepts are defined by cognitive procedures which given a tuple of objects return one of the Boolean values $\{0, 1\}$. For basic concepts, these cognitive procedures are the categorizers associated with perceptually grounded categories. For nonatomic concepts, the cognitive procedures are logical compositions of the cognitive procedures associated with basic concepts. Basic concepts have, therefore, precise and explicit meanings in grounded models, which determine the set of concepts that can constitute the intuitive truths of a grounded model. This can be contrasted with logical theories, in which concepts are defined by simple symbols whose meaning is only constrained by the relationships the axioms in the theory postulate about them.

[1]In logical theories, basic concepts are specified by the non-logical symbols of the language, and defined concepts by mathematical abbreviations or definitions.

It should be observed as well, that the meanings of basic concepts in grounded models are not arbitrary functions from the set of possible tuples of segmented objects on the Boolean values $\{0, 1\}$, as it happens with the meanings of predicate symbols in model theory semantics. They are cognitive procedures which should make intuitively plausible distinctions (in general, relations) using data extracted by realistic sensory channels operating on real world environments. This fact has important consequences on the shape and inferential power of grounded models. First, the meanings of basic concepts are highly constrained by physical aspects of the environment and the sensory-motor apparatus of the agent. Because, in order to qualify as a possible meaning, a cognitive procedure has to be implementable as a relatively easy computation on the range of values produced by realistic sensory channels. Second, the set of intuitive truths is constrained as well by mathematical relationships holding among the meanings of basic concepts. For example, the following relationship holds among the meanings of the concepts $[\text{VPOS } 0.0\ 0.5](obj1)$, $\neg[\text{VPOS } 0.0\ 0.5](obj2)$ and $[\text{VPOS-DIFF } 0.0\ 1.0](obj1, obj2)$.

$$[\text{VPOS } 0.0\ 0.5](obj1) \wedge \neg[\text{VPOS } 0.0\ 0.5](obj2) \rightarrow$$
$$\neg[\text{VPOS-DIFF } 0.0\ 1.0](obj1, obj2)$$

Therefore, the following concept can never be an intuitive truth of a grounded model containing the basic concepts $[\text{VPOS-DIFF } 0.0\ 1.0](obj1, obj2)$ and $[\text{VPOS } 0.0\ 0.5](obj1)$.

$$[\text{VPOS } 0.0\ 0.5](obj1) \wedge \neg[\text{VPOS } 0.0\ 0.5](obj2) \wedge [\text{VPOS-DIFF } 0.0\ 1.0](obj1, obj2)$$

This contrasts with axiomatic theories in which any well formed formula can be stated as an axiom.

The mathematical relationships holding among the meanings of basic concepts are one of the most important features of a grounded model. As we will see later on, they allow agents to reason about their environment without having an explicit axiomatization of it. This form of *intuitive reasoning* is commonly used by people to reason about everyday problems, and it is also the basis of formal reasoning in the sense of mathematics. This is so, because grounded models provide an explanation for the fact that we accept certain axioms as intuitively true when we build axiomatic theories in mathematics.

We can observe a dual character between grounded models and logical theories. Grounded models are constructed around the notion of concept, i.e. the meanings of concepts determine the set of intuitive truths of a grounded model. Logical theories are instead constructed around the notion of axiom, i.e. the set of axioms determine the set of theorems of a theory, and to some extent the meanings of concepts. By looking at our example of grounded model G, one could say that grounded models are logical theories whose axiom set consists only of definitions of concepts. But this is not true, because the meanings of concepts in grounded models are cognitive procedures rather than logical formulas. Sometimes, the behavior of these procedures can be described by mathematical functions, as in the case of grounded model G, but other times it cannot be easily described this way, as in the case of categorizers for approximate concepts. In the first case, intuitive reasoning can be done by exact mathematical methods, as we will see later on, but in the second case different techniques (e.g. simulation) must be applied.

3 Constructing Grounded Models

Grounded models are constructed as a side effect of agents' activity. In particular, the grounded models studied in the paper are constructed by agents as they play discrimination games. A *discrimination game* [Steels, 1996] is played by a single agent. The agent perceives a scene and chooses a topic from the possible tuples of segmented objects in the scene. He then uses his current grounded model to come up with a category or nonatomic concept that is valid for the topic, but not for any other tuple of objects in the context. The game succeeds if the agent can find such a category or concept. If the game succeeds, the use and success counters of the categorizers involved go up[2]. If the game fails, the use counters of the categorizers involved go up, and a repair process in which a new category or concept is generated takes place.

Initially, the agent constructs top-level categorizers for each sensory channel that has contained distinctive data in the recent past. If a channel has the same data for every segment it is not going to be possible to find a distinctive category from it. Afterwards, the agent extends his discrimination trees or constructs new concepts from existing categories or concepts.

A categorizer for a new category is constructed by taking a categorizer node in a discrimination tree and dividing its range into two new subranges. For example, if we take the categorizer for [HPOS 0.0 0.5](x), which is true when the object is in the left most half of a scene, two new categorizers are created by dividing [0.0 0.5] into two halves, one for the range [0.0 0.25] ([HPOS 0.0 0.25](x) or totally left), and one for the range [0.25 0.5] ([HPOS 0.25 0.5](x) or mid left). A new categorizer is added to the tree for each of these halves.

A categorizer for a new concept is constructed by composition of the categorizers of existing categories or concepts. We use three composition operations: negation, conjunction and substitution. These operations allow constructing every free-quantifier formula of the first order language defined by the categories of the grounded model. In general, new concepts are constructed from categories or concepts that have been useful in previous games, i.e. which have a high rate. Two logical compositions are preferred to relate existing categories and concepts: conjunction and implication. Conjunctions tend to be more discriminating than their components, increasing the chances of success in future games. Implications, on the other hand, are commonly used to describe causality, and world facts are often expressed as causal laws.

[2]The use and success counters of a categorizer are used for different purposes, such as computing the categorizer rate (which is the result of dividing success by use), or choosing which categories or concepts should be used to create new concepts.

The final step of the discrimination game, called *assimilation*, is as follows. Whenever the agent constructs a new concept, he checks whether it is an intuitive truth or a falsehood. If the new concept is an intuitive truth, the concept is added to the set of intuitive truths of his grounded model. If it is an intuitive truth and it is already in his grounded model, or if it is a falsehood, the concept is ignored and it is not included in the grounded model. Finally, if the concept is a regular concept, and it is not already in his current grounded model, it is added to the set of defined concepts of the grounded model.

4 Intuitive Reasoning

The process by which an agent tries to determine whether a concept is an intuitive truth, a regular concept or a falsehood of a grounded model is called, in this paper, *intuitive reasoning*. An intuitive truth of a grounded model is a concept whose meaning is true for every possible tuple of segmented objects. A regular concept is a concept whose meaning is true for some tuples of objects, but false for others. And a falsehood is a concept whose meaning is false for every possible tuple of segmented objects.

We hypothesize that intuitive reasoning happens by a process of simulation in natural agents. First, they construct the categorizer of the concept they want to check combining the categorizers of basic concepts of their grounded models. Then, they try to find combinations of values of their sensory channels that can satisfy the categorizer of the concept. This process can be seen as a form of constraint satisfaction by search, in which the agents generate possible combinations of values for sensory channels, and test them using the concept's meaning. Of course, the search cannot be exhaustive, because sensory channels take values on a continuous domain. The search is performed at the level of subregions in which the categorizers involved take different values. Each subregion is represented by a single value of a sensory channel, and the combinations of subregions for sensory channels by tuples of values. As soon as a value is shown incompatible with the concept's meaning or sufficient for satisfying it, all the combinations containing that value can be eliminated from the search space. This allows reducing the complexity of the search process. Sometimes, however, the agents cannot explore the entire space of possibilities, because it is too complex, and they make errors when they try to approximate the result of the search.

If, after the search process, the agents cannot find any combination of values which does not satisfy the concept's meaning, the concept is seen as an intuitive truth. If they find some combinations that satisfy it, and others which do not, the concept is seen as a regular concept; and, otherwise, as an intuitive falsehood.

5 Spatial Reasoning

To clarify the ideas discussed in previous sections, we compare a first order theory T_S, which can be used for reasoning about spatial relations among objects on the plane, with the grounded model G described in section 2. The language of T_S consists of two binary predicates $A(x, y)$ and $R(x, y)$. Its axiom set is as follows.

$$A(x,y) \rightarrow \neg A(y,x) \qquad (1)$$

$$A(x,y) \wedge A(y,z) \rightarrow A(x,z) \qquad (2)$$

$$A(x,y) \wedge A(x,z) \wedge \neg R(y,z) \wedge \neg R(z,y) \wedge y \neq z \rightarrow A(y,z) \vee A(z,y) \qquad (3)$$

$$R(x,y) \rightarrow \neg R(y,x) \qquad (4)$$

$$R(x,y) \wedge R(y,z) \rightarrow R(x,z) \qquad (5)$$

$$R(x,y) \wedge R(x,z) \wedge \neg A(y,z) \wedge \neg A(z,y) \wedge y \neq z \rightarrow R(y,z) \vee R(z,y) \qquad (6)$$

In the classical approach to AI, if one wants to build an agent capable of reasoning about spatial relations, such as *Above* or *Right-of*, one must construct a first order theory like T_S and apply automatic theorem proving methods. Theories like this one are constructed by agent designers, not by the agents themselves, and they must be extended or updated by designers as well when the agents are faced with new challenges. These theories are normally used to determine whether a fact follows or not from the axioms of the theory, i.e. whether it is a theorem, or to extract answers in the form of sets of tuples of objects that satisfy a particular property expressed as a logic formula.

We are going to see how a simple grounded model, such as the one shown in section 2, can be used for the same tasks as T_S. The difference is that grounded models are constructed and updated by agents rather than agent designers, and that inference is done by *intuitive reasoning*.

First, we see that every theorem of T_S is an intuitive truth of the grounded model G. In order to do that, we need to prove that every axiom of T_S is an intuitive truth of G, and that the intuitive truths of G are closed under the inference rule of resolution.

When the behavior of the categorizers for basic concepts can be described by linear constraints, intuitive reasoning can be seen as a process of linear constraint satisfaction. In particular, in the grounded model G, the behavior of the categorizer of every concept can be described by a disjunction of linear constraints which can be computed by replacing every category by a linear constraint describing the behavior of its categorizer in the concept expression, and computing the disjunctive normal form of the result. Proving that the meaning of a concept is true for every possible tuple of segmented objects requires proving that the constraint system associated with the concept is true for every value of every sensory channel. This is equivalent to proving that the constraint system associated with the negation of the concept's meaning is unsatisfiable.

For example, it can be shown that axiom 2 (which states that the relation above – A(x,y) – is transitive) is an intuitive truth of the grounded model G by checking that the following disjunction of constraint systems is unsatisfiable for every value of x, y and z in the interval (0.0 1.0). This formula has been obtained by: (1) replacing every category by a linear constraint describing the behavior of its categorizer in the concept associated with the negation of axiom 2 in the grounded model G; (2) replacing every instance of VPOS(x), VPOS(y) and VPOS(z) by x, y and z in the expression resulting from step 1; and (3) computing the disjunctive normal form of the result of step 2.

$$\{0 < x\text{-}y,\ x\text{-}y < 1.0,\ 0 < y\text{-}z,\ y\text{-}z < 1.0,\ x\text{-}z \leq 0\} \vee$$
$$\{0 < x\text{-}y,\ x\text{-}y < 1.0,\ 0 < y\text{-}z,\ y\text{-}z < 1.0,\ 1.0 \leq x\text{-}z\}$$

It is easy to check that this constraint system is unsatisfiable. A disjunction of constraint systems is unsatisfiable if each disjunct is unsatisfiable. Each disjunct is a linear arithmetic constraint that can be checked by a linear constraint solver, such as the one implemented in Sicstus Prolog.

The rest of the axioms of T_S can be shown to be intuitive truths of the grounded model G by intuitive reasoning as well. Grounded model G provides, therefore, an explanation for the fact that we accept the axioms of T_S as intuitively true, and consequently use them to build logical theories for spatial reasoning.

It can also be proved that intuitive reasoning in grounded models in which the behavior of the categorizers for basic concepts can be described by linear constraints is closed under resolution. That is, if two concepts are intuitive truths of a grounded model, its resolvent is an intuitive truth of the grounded model as well. Therefore, every theorem of T_S can be shown to be an intuitive truth of the grounded model G by intuitive reasoning. Intuitive reasoning is not only much simpler than theorem proving for this domain, but it can also be automated using constraint satisfaction techniques as we have explained above. In fact, the grounded model G can be used to derive many more intuitive truths than the theorems of T_S, because it contains categorizers for a broader set of basic concepts including the standard notions of up(x), down(x), right(x), left(x), dark(x), light(x), darker(x,y) and lighter(x,y).

Grounded models can be used to extract answers as well. For example, we can obtain all the triples of objects in the scene of figure 1 such that $A(x, y) \wedge A(y, z)$ by applying the categorizer of this concept to every triple of segmented objects in the scene, and picking up those triples that satisfy it.

Notice that, as soon as the agents have constructed categorizers for the basic concepts *above* and *right-of*, they are capable of deriving every theorem of T_S by intuitive reasoning. This means that an agent capable of linguistic competence [Steels, 1998] at the level of interpreting and generating first order logic formulas does not need an axiomatization to reach sound conclusions about its environment by intuitive reasoning in this domain. It only needs to understand the basic concepts involved, and the grammar rules by which concept expressions are translated into concept meanings. We have made some recent progress in this aspect, which is illustrated by the following experiment. Figure 2 shows the results of an experiment in which a couple of agents learn both logical categories, which correspond to the usual connectives of propositional logic ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$), and a shared vocabulary to refer to them. The particular language game used to learn logical connectives is a variation of the language games de-

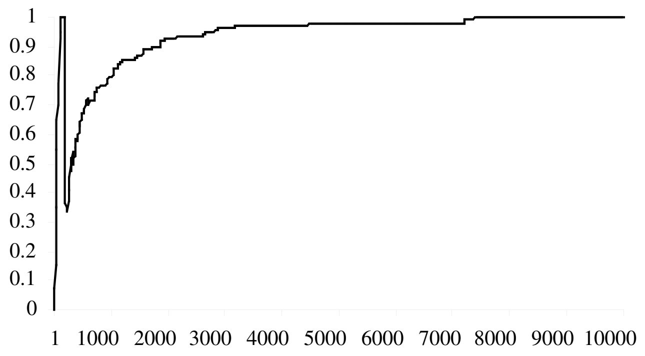

Figure 2: This graph shows the evolution of lexical coherence in a series of 10000 language games. First, the agents learn unary connectives ($\neg$) reaching total coherence very soon. Afterwards they start learning binary connectives ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) reaching 0.97 coherence after 3000 games.

scribed in [Steels, 1999]. The experiment shows that agents can indeed acquire linguistic competence at the level of interpreting and generating free-quantifier first order formulas.

Finally, grounded models can be used for a task that logical theories do not support, namely, *concept generation*. As a side effect of their activity playing discrimination games, the agents construct new concepts which they store and can use afterwards for different purposes, such as classification, discrimination or communication. Some of these new concepts are intuitive truths. These are stored in the set of intuitive truths of a grounded model, and allow the agents to learn general facts about their environment[3]. Others are regular concepts, and are stored in the set of defined concepts of a grounded model. Axiomatizations and conceptualizations get constructed then as a side effect of agents' activity, and do not have to be built into them.

6 Conclusions

We have introduced grounded models and compared them to axiomatic models of mathematics. Grounded models differ from axiomatic theories in establishing explicit connections between language and reality that are learnt through language games. They are constructed and updated by autonomous agents connected to their environment through sensors and actuators using some conceptualization mechanisms and language games described in [Steels, 1999]. They are based on conceptualization and support a form of *intuitive reasoning*, which can be done sometimes by constraint satisfaction and it is argued to be the basis of some axiomatizations. This has been illustrated with a simple example of spatial reasoning.

Acknowledgments

The author would like to thank Luc Steels for many interesting conversations on the topics of the origins of language and grounded representations.

[3]The set of intuitive truths of a grounded model plays an analogous role to that of the axiom set of a logical theory.

References

[Lifschitz, 1993] Vladimir Lifschitz. Circumscription. In Handbook of Logic in Artificial Intelligence and Logic Programming, D. Gabbay and C.J. Hogger, Ed., Oxford University Press, 1993.

[McCarthy, 1959] John McCarthy. Programs with common sense. In *Mechanization of Thought Processes, Proceedings of the Symposium of the National Physics Laboratory*, pages 77–84, 1959.

[McCarthy, 1980] John McCarthy. Circumscription -a form of non-monotonic reasoning. *Artificial Intelligence*,13:27–39, 1980.

[McCarthy, 1986] John McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

[McCarthy, 1990] John McCarthy. *Formalizing Common Sense. Papers by John McCarthy.* Edited by Vladimir Lifschitz. Ablex Publishing Corporation, New Jersey, 1990.

[Piaget, 1985] J Piaget. *The Equilibration of Cognitive Structures: the Central Problem of Intellectual Development.* University of Chicago Press, Chicago, 1985.

[Shoenfield, 1967] John R. Shoenfield. *Mathematical Logic*. Addison-Wesley Publishing Company, 1967.

[Sierra, 2001] Josefina Sierra Santibáñez. Grounded models. In *Working Notes of the AAAI–2001 Spring Symposium on Learning Grounded Representations*, pages 69–74, Stanford University, Stanford, California, March 26–28, 2001.

[Steels, 1996] Luc Steels. Perceptually grounded meaning creation. In *Proceedings of the International Conference on Multi-Agent Systems.* Tokoro, M. (ed.). AAAI Press, Menlo Park Ca. AAAI Press, 1996.

[Steels, 1997] Luc Steels. The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–35, 1997.

[Steels, 1998] Luc Steels. The origins of syntax in visually grounded agents. *Artificial Intelligence*, 103:1–24, 1998.

[Steels, 1999] Luc Steels. *The Talking Heads Experiment. Volume 1. Words and Meanings.* Special pre-edition for LABORATORIUM, Antwerpen, 1999.

[Steels, 2000] Luc Steels. The emergence of grammar in communicating autonomous robotic agents. In *Proceedings of the European Conference on Artificial Intelligence 2000.* Horn, W. (ed.), IOS Publishing, Amsterdam, 2000.

[Steels, 2001] Luc Steels. The role of language in learning grounded representations. In *Working Notes of the AAAI–2001 Spring Symposium on Learning Grounded Representations*, pages 80–85, Stanford University, Stanford, California, March 26–28, 2001.

[Steels and Vogt, 1997] Luc Steels and Paul Vogt. Grounding adaptive language games in robotic agents. In *Proceedings of the European Conference on Artificial Life 97.* The MIT Press, Cambridge Ma, 1997.

[Wittgenstein, 1953] Ludwig Wittgenstein *Philosophical Investigations.* Macmillan, New York, 1953.

Perceptual Anchoring of Symbols for Action

Silvia Coradeschi and **Alessandro Saffiotti**
Center for Applied Autonomous Sensor Systems
Örebro University, S-70182 Örebro, Sweden
http://www.aass.oru.se
silvia.coradeschi@aass.oru.se, alessandro.saffiotti@aass.oru.se

Abstract

Anchoring is the process of creating and maintaining the correspondence between symbols and percepts that refer to the same physical objects. Although this process must necessarily be present in any symbolic reasoning system embedded in a physical environment (e.g., an autonomous robot), the systematic study of anchoring as a clearly separated problem is just in its initial phase. In this paper we focus on the use of symbols in actions and plans and the consequences this has for anchoring. In particular we introduce action properties and partial matching of objects descriptions. We also consider the use of indefinite references in the context of action. The use of our formalism is exemplified in a mobile robotic domain.

1 Introduction

The focus of this paper is the connection between abstract- and physical-level representations of objects in artificial autonomous systems embedded in a physical environment. We call *anchoring* the process of creating, and maintaining over time, this connection.

Anchoring must necessarily occur in any physically embedded system that comprises a symbolic reasoning component. A typical example is the problem of connecting, inside an autonomous robot, a symbol used by a symbolic planner to refer to a physical object to the data in a perceptual system that pertains to the same object. This connection must be dynamic, since the same symbol must be connected to new percepts when the same object is re-acquired. For instance, a robot may be asked to identify and track a specific person in a crowd using visual data and given a linguistic description.

Anchoring is related to *symbol grounding*, defined as the problem of how to give an interpretation to a formal symbol system that is based on something that is not just another symbol system [Harnard, 1990]. Anchoring is an important special case of symbol grounding where the symbols denote individual physical objects.

The recognition of the anchoring problem as a problem *per se* is a recent phenomenon. Although all existing robotics systems that comprise a symbolic reasoning component implicitly incorporate a solution to the anchoring problem, this solution is typical hidden in the code, and it is developed on a system by system basis on a restricted domain. To the best of our knowledge, the first domain independent definition of the anchoring problem was given in [Saffiotti, 1994], while the first attempt at a computational theory of anchoring was reported in [Coradeschi and Saffiotti, 2000]. The goal of this theory was to specify the functionalities and the representations needed to perform anchoring in a general way that is applicable to a large number of systems.

In this paper, we focus on one specific aspect of anchoring: the use of symbols to denote objects in actions and plans, and the anchoring of these symbols to objects in the world. Consider the action 'PickUp(A).' We are interested in the problem of how to anchor the symbol 'A' to the relevant physical object through perception. To do this, we start from the above theory of anchoring, and extend it in three ways. First, we make a distinction between the properties of 'A' which are needed to *identify* the physical object to be used for the action, and those which are needed to *perform* the action. Second, we introduce the concept of *partial matching*, where 'A' can be (tentatively) anchored to an object whenever a required perceptual property cannot be extracted from the sensor data. Third, we consider the use of *indefinite references* in the context of action. Definite references, like "*the* black suitcase,"are meant to refer to one specific object with given properties, while indefinite ones, like "*a* black suitcase" are meant to refer to an arbitrary object in a given class [Russell, 1905].

In order to clarify the use of anchoring, we show two experiments performed on a real robot. They illustrate how anchoring can be integrated in a robot architecture, and how its functionalities can be used to connect the symbols used by a planner to the data acquired by a vision system. The examples also show the essential difference between two ways to treat actions that involve an indefinite reference. This reference can be resolved by the symbol system (planner), or by the anchoring process.

2 A Basic Model of Anchoring

We summarize here the basic elements of the computational theory of anchoring defined in [Coradeschi and Saffiotti, 2000]. The theory considers an agent that includes a symbol system and a perceptual system, and it focuses on the problem of creating and maintaining a correspondence between

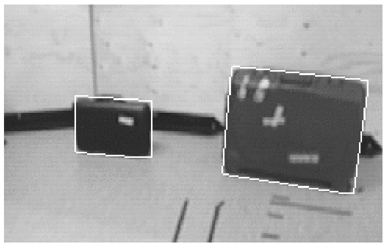

Figure 1: Two percepts extracted from a camera image.

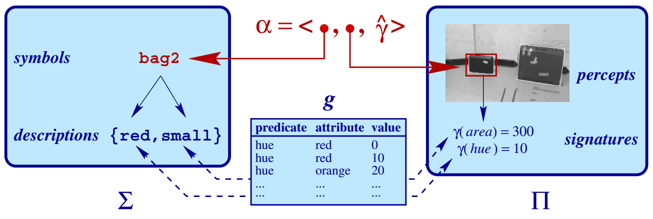

Figure 2: The elements of anchoring.

symbols and percepts that refer to the same physical object. It consists of a static part and a dynamic part. The static part includes the following.

- A *symbol system* Σ including: a set $\mathcal{X} = \{x_1, x_2, \ldots\}$ of individual symbols (variables and constants); a set $\mathcal{P} = \{p_1, p_2, \ldots\}$ of predicate symbols; and an inference mechanism whose details are not relevant here.
- A *perceptual system* Ξ including: a set $\Pi = \{\pi_1, \pi_2, \ldots\}$ of percepts; a set $\Phi = \{\phi_1, \phi_2, \ldots\}$ of attributes; and perceptual routines whose details are not relevant here. A percept is a structured collection of measurements assumed to originate from the same physical object; an attribute ϕ_i is a measurable property of percepts, with values in the domain D_i. We let $D = \bigcup_i D_i$.
- A *predicate grounding relation* $g \subseteq \mathcal{P} \times \Phi \times D$, that embodies the correspondence between unary predicates and values of measurable attributes.

Example. Σ may be a planner that includes the individual symbol 'A' and the predicate symbols 'large' and 'small.' Ξ may be a vision system able to recognize suitcases: from the image shown in Fig. 1, Ξ may extract two percepts π_1 and π_2. Attributes computed by Ξ may include 'color' and 'width.' The predicate grounding relation g may include the triple $\langle\text{small, width, }10\rangle$: this says that the measure 10 for an object's observed width is consistent with the predication of its being small.[1]

The g relation concerns properties, but anchoring concerns objects. The following definitions allow us to characterize objects in terms of their (symbolic and perceptual) properties.

Definition 1 *A* symbolic description $\sigma \in 2^{\mathcal{P}}$ *is a set of unary predicates.*

Definition 2 *A* perceptual signature $\gamma : \Phi \to D$ *is a partial function from attributes to attribute values. The set of attributes on which γ is defined is denoted by feat(γ). $\Gamma = (\Phi \to D)$ is the set of all γ.*

Intuitively, a symbolic description lists the predicates that are considered relevant to the perceptual recognition of an object; and a perceptual signature gives the values of the measured attributes of a percept (and it is undefined for the remaining ones). The g relation can then be used to define a function *match*(σ, γ) that says whether or not the values in the perceptual signature γ are consistent with a given symbolic description σ.

[1]For the sake of simplicity we consider here a very simple g. The g relation can be quite complex in real domains.

The dynamic part of the model tells us, at any time t, what properties are associated to symbols in Σ, and what attribute values are associated to percepts in Ξ.

- A *description state* $DS_t : \mathcal{X} \to 2^{\mathcal{P}}$ that associates each individual x with its symbolic description at time t.
- A *perceptual state* $PS_t : \Pi \to \Gamma$ that associates each percept $\pi \in \Pi$ to its perceptual signature at time t. If π is not perceived at time t, then $PS_t(\pi)$ is everywhere undefined. The set of percepts which are perceived at t is denoted by V_t.

Example. Consider our previous example. At time t, the symbol system may associate property 'small' to symbol 'A' by having $\texttt{small} \in DS_t(A)$. The perceptual system may extract the width of the two percepts in the image, and associate them with the perceptual signatures $PS_t(\pi_1) = \gamma_1$ and $PS_t(\pi_2) = \gamma_2$ such that $\gamma_1(\text{width}) = 10$ and $\gamma_2(\text{width}) = 20$.

The role of anchoring is to establish a correspondence between a symbol x used in the symbol system to denote an *object* in the world, and a percept π generated in the perceptual system by the same object. This is done by comparing the symbolic description $DS_t(x)$ and the perceptual signature $PS_t(\pi)$ via the *match* function, hence via the g grounding relation. In the previous example, the g relation includes the triple $\langle\text{small, width, }10\rangle$, therefore the description of 'A' and the perceptual signature of π_1 can be matched, thus suggesting that the symbol 'A' might be anchored to the percept π_1.

The above correspondence is reified in an internal data structure α, called *anchor*. Since new percepts are generated continuously within the perceptual system, this correspondence is indexed by time.

Definition 3 *An* anchor α *is any partial function from time to triples in $\mathcal{X} \times \Pi \times \Gamma$.*

At every moment t, $\alpha(t)$ contains: a symbol, meant to denote an object inside Σ; a percept, generated inside Ξ by observing that object; and a signature, meant to provide the (best) estimate of the values of the observable properties of the object. We denote these components by α_t^{sym}, α_t^{per}, and α_t^{sig}, respectively. If the object is not observed at time t, then α_t^{per} is the 'null' percept $\perp$, and α_t^{sig} still contains the best available estimate. Intuitively, an anchor can be seen as an internal, sensori-motor level representation of a physical object — see Fig. 2.

In order for an anchor to satisfy its intended meaning, the symbol and the percept in it should refer to the same physical object. This requirement cannot be formally stated inside the system. What can be stated is the following.

Definition 4 *An anchor* α *is* grounded *at time t iff* $\alpha_t^{\text{per}} \in V_t$.

We informally say that an anchor α is *referentially correct* if, whenever α is grounded at t, then the physical object denoted by α_t^{sym} is the same as the one that generates the perception α_t^{per}. The *anchoring problem*, then, is the problem to find referentially correct anchors.

3 Extending the Model

The above model provides the basic ingredients of a general theory of anchoring, with no assumption as to the task for which anchoring is performed. In this paper, however, we focus on the use of anchoring to connect symbols for actions to physical objects through perception. From this perspective, the anchoring process should make sure that the anchor: (i) represents a physical object which has the intended properties for the intended action, and (ii) contains information about the perceptual properties which are needed in order to perform the action. For instance, if the action is meant to pick up a green suitcase, then the anchor should represent an object which is a green suitcase, and its signature should include an estimate of the position and orientation of this suitcase.

3.1 Action properties

The *match* function makes sure that the anchor is adequate with respect to the properties stored in the description state, i.e., $DS_t(A)$. In order to make sure that the anchor's signature also contains the properties that are relevant for action, we extend the dynamic part of our model by including the following.

- An *action parameter state* $A_t : \mathcal{X} \to 2^{\mathcal{P}}$ that associates each individual x with the set of properties that need to be known in order to act on the object denoted by x.

In our example, A_t(A) would specify the position and orientation of the suitcase. In the following we call *matching properties* the set of predicates in $DS_t(A)$ and *action properties* the set of predicates in $A_t(A)$

Note that the predicates in A_t are not used in the matching process: these predicates only indicate that the corresponding attributes must be included in the anchor's perceptual signature γ. However, these predicates can be used as *default assumptions* about the expected properties of the object in order to start action before the object is actually perceived. For instance, we may have an expectation about the position of a suitcase: this expectation can be included in the signature of the anchor in order to start approaching that position until the actual suitcase is perceived. These expectations can also be used to focus the perceptual system.

Having a property in the action state or in the description state may affect the meaning of an action. In our "PickUp(A)" example, if the position is not included in the description state, then 'A' will be anchored to any green suitcase, irrespective of its position. If the position of a given suitcase in included in the description state, then 'A' will only be anchored to the specific suitcase at that position. The first setup encodes the action "pick up *a* green suitcase," while the second one encodes the action "pick up *the* green suitcase at the given position" [Saffiotti, 1994]. Note that the object of the action is denoted by an indefinite reference in the first case, and by a definite one in the second case.

3.2 Partial Matching

Some actions may affect the perceptual information which is gathered by the agent: for instance, moving closer to an object may allow the perceptual system to observe more properties of the object. For example, suppose that we are interested in a suitcase with a white label on it: depending on the distance and angle of the suitcase, the label may not be visible.

Recognizing the fact that not all attributes of a percept may be extracted at all times brings about the need to redefine the meaning of the *match* function: $match(\sigma, \gamma)$ should check that the signature γ is consistent with the descriptor σ for those attributes which have actually been observed, but it should ignore the ones which have not been observed. The following is a possible way to define the *match* function.

$$match(\sigma,\gamma) = \begin{cases} \emptyset \text{ if } \exists p \in \sigma.\,(obs(p,\gamma) \wedge \neg cons(p,\gamma)) \\ \{p \in \sigma \mid obs(p,\gamma)\} \text{ otherwise} \end{cases}$$

where

$$\begin{aligned} obs(p,\gamma) &\Leftrightarrow \exists \phi \in feat(\gamma).\exists d \in D.g(p,\phi,d) \\ cons(p,\gamma) &\Leftrightarrow \exists \phi \in feat(\gamma).g(p,\phi,\gamma(\phi)) \end{aligned}$$

Intuitively, $obs(p, \gamma)$ says that an attribute related to p (via the g relation) has been observed in γ, and $cons(p, \gamma)$ says that the observations in γ are consistent with the p predicate according to g. $match(\sigma, \gamma)$ returns $\emptyset$ if there was a mismatch between some predicate in σ and the observed values; otherwise it returns the subset of the predicates in σ for which a (consistent) value has actually been observed.

It is useful to keep track of which of the predicates in the symbolic description for a symbol have actually been observed, and which ones have not. To do this, we extend the definition of an anchor to include a list of the observed predicates as follows.

Definition 3 (bis) *An* anchor *is any partial function from time to tuples in* $\mathcal{X} \times \Pi \times \Gamma \times 2^{\mathcal{P}}$.

We write α^{obs} to denote the fourth element of an anchor $\alpha(t)$.

4 The Functionalities of Anchoring

In order to turn the above model into a useful computational framework, we need to define which functionalities are needed in order to solve the anchoring problem for a given symbol x. In [Coradeschi and Saffiotti, 2000] three main functionalities have been identified: (i) to create a grounded anchor the first time that the object denoted by x is perceived; (ii) to update the anchor when we need to reacquire the object after some time that it has not been observed; and (iii) to continuously update the anchor while observing the object. Given the above extensions to the model, these functionalities can be defined as follows. (t denotes the time at which the functionality is called.)

Find Take a symbol x and return a grounded anchor defined at t, and undefined elsewhere. In case of multiple matching percepts, return one anchor for each of them. This is summarized by the following pseudo-code.

```
procedure Find $(x, t)$
  $\Pi \leftarrow \{\pi \in V_t \mid \mathit{match}(\mathit{DS}_t(x), \mathit{PS}_t(\pi)) \neq \emptyset\}$
  if $\Pi = \emptyset$
    then fail
    else for $\pi_i \in \{\pi_1, \ldots, \pi_n\} = \Pi$
            $\mu_i \leftarrow \mathit{match}(\mathit{DS}_t(x), \mathit{PS}_t(\pi_i))$
            $\gamma_i \leftarrow \mathit{Attributes}(A_t(x), \mathit{PS}_t(\pi_i), \mu_i)$
            $\alpha_i(t) \leftarrow \langle x, \pi_i, \gamma_i, \mu_i \rangle$
return $\{\alpha_1, \ldots, \alpha_n\}$
```

The *Attributes* function returns the part of the perceptual signature $\mathit{PS}_t(\pi_i)$ that only includes the "interesting" attributes, that is, those that correspond to either description properties or to action properties. (A specific implementation may also include attributes which are needed by the perceptual system to track the object, e.g., its position and velocity.)

Reacquire This function is used to find an object when there is a previous perceptual experience of it. Take an anchor α defined at time $t - k$ and extend α's definition to t. First, predict a new signature γ; then see if there is some new percept that is compatible with the prediction and the symbolic description; in case of multiple matching percepts, use a domain dependent selection function. If one percept is found, update γ. Prediction, verification of compatibility, and updating are domain dependent; verification should typically use *match*.

```
procedure Reacquire $(\alpha, t)$
  $x \leftarrow \alpha^{\text{sym}}_{t-k}$
  $\mu \leftarrow \emptyset$
  $\gamma \leftarrow \text{Predict}(\alpha^{\text{sig}}_{t-k}, x, t)$
  $\pi \leftarrow \text{Select}\{\pi' \in V_t \mid \text{Verify}(\mathit{DS}_t(x), \mathit{PS}_t(\pi'), \gamma) \neq \emptyset\}$
  if $\pi \neq \perp$ then $\mu \leftarrow \mathit{Verify}(\mathit{DS}_t(x), \mathit{PS}_t(\pi_i), \gamma)$
                  $\gamma \leftarrow \text{Update}(\gamma, \mathit{PS}_t(\pi), \mathit{DS}_t(x))$
  $\alpha(t) \leftarrow \langle x, \pi, \gamma, \mu \rangle$
return $\alpha$
```

If Reacquire fails to find a matching percept, then $\alpha(t)$ contains the predicted signature and the 'null' percept $\perp$. Note that in this case $\alpha(t)$ is not grounded.

Track Take an anchor α defined for $t - 1$ and extend its definition to t. It is used in the special case of reacquisition whenever the object is kept under constant observation. Prediction is in general much simpler than in the Reacquire case, and verification is only made with respect to the previously perceived attributes via a domain dependent function MatchSignature. This functionality could for instance be implemented with a Kalman filter.

```
procedure Track $(\alpha)$
  $x \leftarrow \alpha^{\text{sym}}_{t-1}$
  $\gamma \leftarrow \text{OneStepPredict}(\alpha^{\text{sig}}_{t-1}, x)$
  $\pi \leftarrow \text{Select}\{\pi' \in V_t \mid \text{MatchSignature}(\gamma, \mathit{PS}_t(\pi')) \neq \emptyset\}$
  if $\pi \neq \perp$ then $\gamma \leftarrow \text{Update}(\gamma, \mathit{PS}_t(\pi), x)$
```

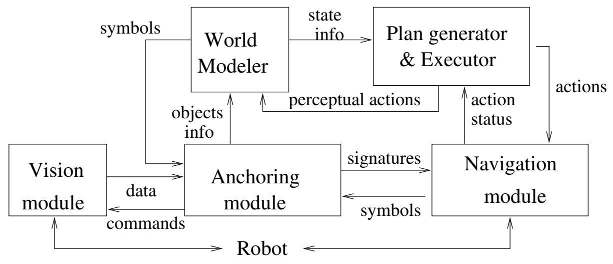

Figure 3: The robot architecture used in our examples.

```
  $\alpha(t) \leftarrow \langle x, \pi, \gamma, \alpha^{\text{obs}} \rangle$
return $\alpha$
```

Notice that the anchor(s) computed always include the best current estimate of the observable properties needed to perform the actions (γ), and an indication of which parts of the symbolic description have actually been observed (μ). The next section will show two examples of use of anchoring that further clarify the role of this information.

5 Examples

In this section we present two examples of anchoring with indefinite references implemented in a robotic system. The robot, a Nomad 200, uses sonars for navigation and vision data to identify objects in the environment. The two-layered decision making architecture of the robot is shown in Fig. 3.

The higher layer includes a plan generator (ETLplan), a plan executor, and a world modeler. ETLplan is a conditional planner capable of generating plans with perceptual actions and conditional branches [Karlsson, 2001]. The plan executor checks the conditions and invokes the actions in the plans generated by the planner. The world modeler maintains information about objects and places relevant for the task. It can obtain additional information about objects from the anchoring module.

The lower layer includes a navigation and a vision module. The navigation module is a simplified version of the fuzzy behavior-based controller defined in [Saffiotti *et al.*, 1995], which executes the actions sent by the plan executor. An example of an action is `(gonear A)`. The vision module contains vision routines for recognizing objects and for calculating properties such as color and size.

The anchoring module provides the connection between the symbols used by the planner and the world modeler, and the perceptual data provided by the vision module and used by the navigation module. It receives requests to create and update anchors and it provides the relevant information about the anchored symbols to the world modeler. It is in this module that the g function is encoded. In addition, the navigation module uses the symbols that constitute the arguments to its actions when requesting information from the anchoring module about the corresponding objects. For instance while executing the action `(gonear A)`, it regularly requests the position of `A`. Finally, the anchoring module controls the vision processing, activating routines for recognizing objects and providing parameters (e.g., expected position) to focus perceptual attention.

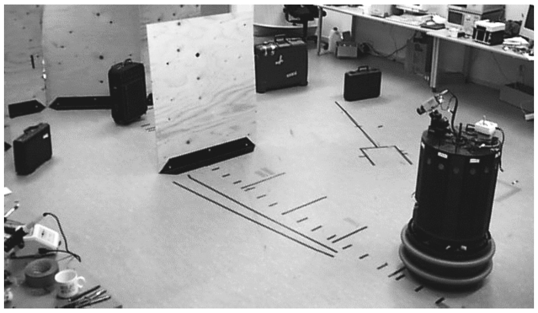

Figure 4: The initial setup in the first example.

5.1 Anchoring with partial matching

This example has the aim to show how the anchoring module works in practice and in particular the use of partial matching. It also serves to illustrate one way to handle indefinite references. The robot has the task to “find a black suitcase with a white mark and to go near it”. The initial scenario is shown in Fig. 4. The largest suitcase is green, while the other three suitcases are black. The world model contains information about three objects: the two small black suitcases, identified in the world model by the symbols C (the one on the right) and B (the one on the left), and the green suitcase identified by the symbol A.

The goal given to the plan generator has the form `(exists (?x) (and (suitcase ?x) (black ?x) (white-mark ?x) (near ?x)))`, that is, the goal is to be near to a black suitcase with a white mark. The world modeler contains the information that both B and C are black suitcases, but does not have information about the mark. Therefore, the plan generator creates a plan that consists of first going near suitcase C and looking for the mark. If the mark is found, the execution stops with success. Otherwise, the robot goes near to suitcase B to look for the mark. Note that the original indefinite reference (“a black suitcase with a white mark”) has now been turned onto two alternative definite references B and C. The actual plan is as follows:

```
((gonear C) (observe C)
 (if ((white-mark C . true)) (:success))
 (if ((white-mark C . false))
     ((gonear B) (observe B)
      (if ((white-mark B . true)) (:success))
      (if ((white-mark B . false)) (:fail)))))
```

Fig. 5 (left) schematically indicates the path followed by the robot during plan execution. The robot goes first to suitcase C, checks the mark (`(observe C)`), does not find it, and goes to suitcase B where it successfully identifies a mark.

The navigation module, while executing `(gonear C)`, is regularly requesting the anchoring module to keep C anchored. The latter uses the track functionality to keep track of C while the robot is moving.

The matching properties provided to the anchoring module are the properties attached to the symbol `C`, such as color, position, and shape, and the fact that the suitcase should have a white mark. The vision routines, due to the distance between the camera and the suitcase, cannot discriminate if there is a mark on suitcase C. A naive matching function might reject this suitcase, given that one property is not satisfied. Partial matching allows us to consider the case that the property is actually not perceivable, and the suitcase is still anchored.

A final observation is that while the robot is moving towards suitcase B, a third black suitcase, which was previously occluded, is also perceived. However, this suitcase is ignored because it does not match the perceptual signature (incl. position) of suitcase B.

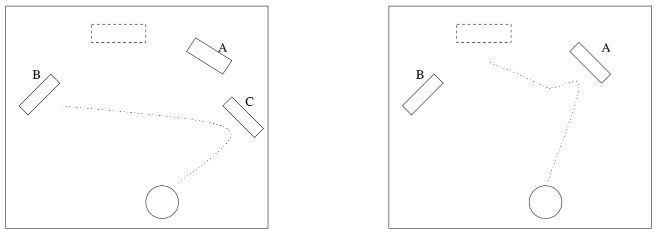

Figure 5: An illustration of the path of the robot in the first (left) and second (right) example.

5.2 Anchoring with indefinite references

This example shows the handling of indefinite references at the level of the anchoring module, as opposed to at the higher symbolic level in the previous example.

The task is to “go near a green suitcase and then go near a black suitcase”. The world model contains initially two suitcases, one green denoted by the symbol A and one black denoted by B. In this case however the plan is to go near a generic object that has the properties of being a green suitcase and then to go near an object that has the properties of being a black suitcase, `((gonear x) (gonear y))`.[2] The difference is that the symbols used for action now do not denote fixed specific objects, but any object in the class of objects satisfying the given matching properties. In case of `x`, these are black and suitcase. The positions of the two known objects are used as action properties, that is, they are used as an initial value for the `gonear` action. However they are not considered to be matching properties.

The path of the robot is illustrated in Fig. 5 (right). The robot first moves successfully near the green suitcase. However, when it turns towards the black suitcase, this suitcase has been removed. The robot starts moving toward the recorded position of the black suitcase, as its position was given as action property. While the robot moves, a previously occluded black suitcase is perceived: the dotted suitcase in the figure. This suitcase matches the required properties as it is black. It is therefore anchored and the robot goes near it. The fact that the position and the perceptual signature of this second black suitcase is different from those of the first one does not constitute a problem; the anchoring module just matches the percepts with the properties present in the description state, even if it uses the expected position of the first black suitcase to direct the perception and to provide information to the navigation module.

[2]This plan, and the corresponding symbolic descriptors for `x` and `y`, have been created by hand, because existing planning systems only consider specific, previously known objects.

6 Discussion

The autonomous robotics and machine vision literature contains a few examples in which the need and the role of anchoring, under different names, has been explicitly identified, e.g., [Hexmoor *et al.*, 1993], [Saffiotti *et al.*, 1995], [Jung and Zelinsky, 2000], [Chella *et al.*, 1998], [Bajcsy and Košecká, 1994], [Satoh *et al.*, 1997], [Horswill, 1997], [Wasson *et al.*, 1999]. However, all the works above describe specific implementations and do not attempt a study of the general concept of anchoring.

To our knowledge, [Coradeschi and Saffiotti, 2000] was the first attempt to state the general anchoring problem in formal terms. The main technical contribution of this paper is the extension of the above framework with the introduction of two elements needed to deal with the anchoring of symbols in actions and plans. First we have introduced action properties, that is properties that are used in the execution of an action, but are not relevant for finding the correct object. Second, we have introduced partial matching to be able to distinguish between properties that are not satisfied and properties that are not presently available for perception, that is one cannot presently determine whether they are satisfied or not. As we consider anchoring from an action perspective we can deal with the latter case actively, for instance by observing the object from a better position. Anchoring in the presence of partial matching is actually tentative: if a property previously not observed is observed at a later point and it is discovered not to match the description, the anchor can be removed.

The problem of connecting linguistic descriptions of objects to their physical referents has been largely studied in the philosophical and linguistic tradition. These traditions provide a rich source of inspiration for the conceptualization of the anchoring problem. For instance [Russell, 1905] made a distinction between definite and indefinite references. To this respect our examples show two different ways to resolve an indefinite reference in the context of an action. In the first example, the planner resolves the indefinite reference (*a* black suitcase with a mark) by instantiating a variable to known matching objects in the world model, and then it uses definite references (B and C) in its actions. Note that in order to handle the appearance of a new suitcase, it would be necessary to re-plan. In the second example, the plan contains indefinite references, which are sent as such to the anchoring module. Instantiation is delegated to the anchoring module, which is able to shift "on the fly" to the new suitcase. Note that resolution of ambiguities might be more difficult in this case, without the support of the high level reasoning.

The first approach seems to be suitable for relatively static domains where there is the time to re-plan, while the second approach seems to be more appropriate in highly dynamic domains. In fact, we have used a similar approach in the Sony AIBO robots in the RoboCup competition [Saffiotti and LeBlanc, 2000]. This domain is highly dynamic, but there are few ambiguities with respect of anchoring of objects.

Acknowledgements This work was funded by the Swedish KK Foundation. We thank Lars Karlsson and Zbigniew Wasik for providing substantial help in running the robot experiments.

References

[Bajcsy and Košecká, 1994] R. Bajcsy and J. Košecká. The problem of signal and symbol integration: a study of cooperative mobile autonomous agent behaviors. In *Proceedings of KI-95*, LNCS, pages 49–64, Berlin, Germany, 1994. Springer.

[Chella *et al.*, 1998] A. Chella, M. Frixione, and S. Gaglio. An architecture for autonomous agents exploiting conceptual representations. *Robotics and Autonomous Systems*, 25:231–240, 1998.

[Coradeschi and Saffiotti, 2000] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: preliminary report. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 129–135, Austin, 2000.

[Harnard, 1990] S. Harnard. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[Hexmoor *et al.*, 1993] H. Hexmoor, J. Lammens, and S. C. Shapiro. Embodiment in GLAIR: A grounded layered architecture with integrated reasoning for autonomous agents. In *Proc. of the Florida AI Research Sympos.*, pages 325–329, 1993.

[Horswill, 1997] I. Horswill. Visual architecture and cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2):277–292, 1997.

[Jung and Zelinsky, 2000] D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots journal*, 8(3), 2000.

[Karlsson, 2001] Lars Karlsson. Conditional progressive planning under uncertainty. In *Proc. of IJCAI-01*, 2001.

[Russell, 1905] B. Russell. On denoting. In *Mind XIV*, pages 479–493. 1905.

[Saffiotti and LeBlanc, 2000] A. Saffiotti and K. LeBlanc. Active perceptual anchoring of robot behavior in a dynamic environment. In *IEEE Int. Conf. on Robotics and Automation*, 2000.

[Saffiotti *et al.*, 1995] A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995.

[Saffiotti, 1994] A. Saffiotti. Pick-up what? In C. Bäckström and E. Sandewall, editors, *Current trends in AI Planning*, pages 266–277. IOS Press, Amsterdam, Netherlands, 1994.

[Satoh *et al.*, 1997] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in video by the integration of image and natural language processing. In *Proc. of IJCAI-97*, pages 1488–1493, 1997.

[Wasson *et al.*, 1999] G. Wasson, D. Kortenkamp, and E. Huber. Integrating active perception with an autonomous robot architecture. *Robotics and Autonomous Systems*, 26:175–186, 1999.