Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2015), *29*, 249–256.
© Cambridge University Press 2015 0890-0604/15
doi:10.1017/S0890060415000207

On the computability of affordances as relations

JONATHAN R.A. MAIER
General Engineering Program, Clemson University, Clemson, South Carolina, USA

(Received March 1, 2014; Accepted February 1, 2015)

Abstract

One of the principal advantages of affordance-based design is that Gibson's theory of affordances is a relational theory, akin to other relational approaches such as relational biology and relational computer science. The relationships between artifacts and their designers and users are of such primary importance that only a theory that is able to deal with those relationships directly appears to be sufficient for describing the wide breadth of problems in engineering design. However, there is no precise definition for what qualifies as a relational theory. In mathematics, we do find something approaching a theory of relations, dating back at least to Charles Peirce's *Logic of Relatives* around 1870. While rather general, Peirce's ideas on the subject laid the foundation for advances in the 20th century, including the relational model of databases. This paper is a first attempt at applying the mathematics of relations to affordances, with the aim of more precisely characterizing affordances, which heretofore have been difficult to define and, lacking appropriate mathematics, nearly impossible to subject to computation. Meanwhile, the implicit computability of affordances as relations is demonstrated by examples drawn from engineering, physics, computer science, biology, and architecture.

Keywords: Affordance; Computability; Relations

1. INTRODUCTION

One of the principal advantages of affordance-based design is that Gibson's theory of affordances is a relational theory, akin to other relational approaches such as relational biology and relational computer science. In previous work, my colleagues and I have developed a generalized theory of affordances (Maier & Fadel, 2009*a*), as well as several methods for design (Maier & Fadel, 2009*b*); however, our emphasis has never been on design automation, which would require a computational formalism. One proposal that we made, for the use of design exemplars to model affordances, has not yet been implemented, and remains as future work (Maier, Anandan, et al., 2007). Our generalized theory of affordances rests upon several important properties of affordance that any mathematical or computational approach to manipulating affordances should also support. All of these properties follow from Gibson's final discussion of affordance (Gibson, 1979, p. 127–137), which is worth repeating here (all emphases are his):

> The *affordances* of the environment are what it *offers* the animal, what it *provides* or *furnishes*, either for good or ill. The verb *to afford* is found in the dictionary, but the noun *affordance* is not. I have made it up. I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment.
>
> As an affordance . . . for a species of animal, however, they have to be measured *relative to the animal*. They are unique for that animal. They are not just abstract physical properties. . . . An affordance points both ways, to the environment and to the observer. . . . affordances are properties of things taken with reference to an observer.

The five properties of generalized affordances can be summarized as follows:

1. *Complementarity:* An affordance depends on at least two interacting subsystems (e.g., a user *and* an artifact). This follows Gibson's original discussion that affordances are not properties of any one thing in isolation (Gibson, 1979, pp. 127–128). This point has been reinforced in depth by Stoffregen (2003).
2. *Polarity:* Affordances can be positive or negative, for good or for ill, beneficial or harmful. This again follows from Gibson's original discussion (Gibson, 1979, pp. 127–137).

Reprint requests to: Jonathan R.A. Maier, General Engineering Program, 106A Holtzendorff Hall, Clemson University, Clemson, SC 29634, USA. E-mail: jmaier@clemson.edu

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

3. *Multiplicity:* Two interacting subsystems can have multiple affordances. This again follows from Gibson's original discussion, which cites multiple affordances between air and an animal, and between animals themselves, among other examples (Gibson, 1979, pp. 127–128).
4. *Quality:* There is an opportunity to describe how well one subsystem affords something to another subsystem. Gibson intimates this in that affordances can be "measured" (Gibson, 1979, p. 127).
5. *Form dependence:* The structure of the interacting subsystems determines what affordances exist, as well their quality. They are not purely abstract, conceptual, or metaphysical. Gibson does not state this explicitly; he seems somewhat undecided about it ("It is both physical and psychical, yet neither. . . . They are neither physical nor phenomenal"; Gibson, 1979, pp. 127–128).[1] However, all of the examples Gibson lists of subsystems that interact to create affordances all have physical structure, such as terrestrial surfaces, other animals, air, water, and solids. Structure does not have to be entirely physical; for example, in previous work I have suggested analyzing the structure of software with respect to affordances (Maier, 2006). In all cases, a change of structure implies a change in affordances and, consequently, a change in behavior. I regard the property of form-dependence as being essential to design, and have leveraged it to build such methodological tools as the affordance structure matrix (see, e.g., Maier, Ezhilan, et al., 2007).

Meanwhile, the computation of affordances has been approached in various ways by other researchers. For example, within the realm of perceptual psychology, several affordances have been measured, such as stair climbability by Warren (1984). Precise formal definitions for the concept of affordance have been offered by various authors, notably Turvey (1992) and then Stoffregen (2003), although these definitions do not appear to have been implemented or manipulated computationally.

In the field of robotics, situated robotic systems as envisioned early on by Brooks (1991) have been programmed to recognize and make use of affordances (see, e.g., Murphy, 1999). Progress in that field has progressed remarkably, as reported, for example, in Rome et al. (2006), related to the European MACS project ("Multi-Sensory Autonomous Cognitive Systems Interacting With Dynamic Environments for Perceiving and Using Affordances"). Roboticists have clearly begun to manipulate *specific* affordances computationally. However, there does not appear to be any literature yet on computational support for affordances *in general*. Such is the aim of this paper.

[1] An excellent review of Gibson's evolving view of the concept of affordance over his career has been provided by Jones (2003).

2. AN APPROACH TO COMPUTING AFFORDANCES BASED ON THE NOTATION OF PEIRCE

2.1. Background

Contemporary students of computer science will no doubt be familiar with De Morgan's laws and Boolean logic, which have proved to be so useful in programming modern computers that work, at their most basic level, with binary mathematics.

In the 19th century when De Morgan and Boole made their important contributions to the field of mathematical logic, they seem to have been primarily concerned with the application of their math to issues of philosophy, for example, how to test if a given proposition is true or false with respect to other propositions. They had no way to know, of course, of the eventual application to digital computers.

Immediately following the pioneering work of 19th-century logicians De Morgan and Boole, another intellectual of towering abilities, Charles Peirce, extended their logic from the somewhat simple realm of absolute terms such as true and false to the realm of relations (Peirce, 1873).

Although De Morgan's and Boole's systems of logic found more fundamental application to computer science, Peirce's approach found application in the digital age as well, in particular to relational databases, of which Internet search engines are an offshoot.

As mentioned in Section 1, we now appear to be in a similar position in engineering design, where absolute terms such as constraints and other forms of traditional requirements capture are not always adequate to represent the myriad of relationships between designers and users, and the artifacts they interact with over time.

Thus, it will be beneficial to review the logical system of Peirce in the remainder of this section. In the following section, Peirce's logic will be applied to affordances in particular.

2.2. Peirce's notation

Peirce introduces three classes of terms, absolute terms, relative terms, and conjugate terms, as summarized in Table 1. For the purposes of this paper, I have added some examples to each category, such as gear, proximity, and affordance.

Peirce proceeds to show that normal arithmetic operations that apply to real numbers likewise apply to absolute terms

Table 1. *Categories in Peirce's logic of relatives*

Absolute Terms	Simple Relative Terms	Conjugate Terms
Horse	Father of –	Giver of – to –
Tree	Lover of –	Buyer of – for – from –
Man	Servant of –	Proximity between – and –
Gear	Triplet of –	Affordance of – between – and –

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

(as being natural numbers), and more important, apply to relatives. Peirce proceeds to show that several mathematical theorems likewise make sense when the variables are considered as relations. These theorems include the Taylor series and the MacLaurin series and the binomial theorem, all of which are beyond the scope of this paper. Thus, let us consider something simpler: what Peirce calls involution. A convenient property of exponentials involving positive real numbers is that

$$(\mathbf{x}^{\mathbf{a}})^{\mathbf{b}} = \mathbf{x}^{(\mathbf{a}\,\mathbf{b})}.$$

For example, if $\mathbf{x} = 2$, $\mathbf{a} = 3$, $\mathbf{b} = 4$, then

$$(2^3)^4 = 2^{(3\times 4)} = 4096.$$

Peirce shows that logically the same holds for relatives (as well as absolute terms such as the positive real numbers), by taking the meaning of $\mathbf{x}^{\mathbf{a}}$ as "everything which is an $\mathbf{x}$ for every individual of $\mathbf{a}$," and the meaning of $\mathbf{a}\ \mathbf{b}$ as the "relation of $\mathbf{a}$ to $\mathbf{b}$." For example, if

$\mathbf{x}$ = slots
$\mathbf{a}$ = shafts
$\mathbf{b}$ = gears

then,

$$\mathbf{a}\ \mathbf{b}$$

means "relation of shafts to gears."

REMARK. An engineer would be trained more precisely as to what this relation means, considering, for example, how gears are mounted to shafts, and how shafts are involved in power transmission between gears and other machine elements. The notation $\mathbf{a}\ \mathbf{b}$ captures this important relationship in abstract form that can be mathematically, that is, computationally, manipulated. Note that in this case it appears $\mathbf{a}\ \mathbf{b} = \mathbf{b}\ \mathbf{a}$, but this is not true in general. ■

Similarly,

$$\mathbf{x}^{\mathrm{a}}$$

means "everything which is a slot for every individual shaft."

REMARK. The set of shafts would usually be a finite set, such as those capable of being produced by available machine tools, or available from some catalog. ■

In addition,

$$\mathbf{x}^{\mathrm{b}}$$

means "everything which is a slot for every individual gear."

REMARK. Again an engineer might consider this latter relationship to capture the set of all possible slots, that is, keyways (varying perhaps in shape and or depth), that would fit a particular set of gears. Something similar could be pictured for the former relationship between slots and shafts. ■

Finally,

$$(\mathbf{x}^{\mathbf{a}})^{\mathbf{b}} = \mathbf{x}^{(\mathbf{a}\,\mathbf{b})}$$

formally means that "everything which is a (everything which is a slot for every individual shaft) for every gear" is the same as "everything which is a slot for every (relation of shafts to gears)." More simply, both sides of the identity simply refer to all possible slots for all possible shafts for a given set of gears.

REMARK. Although typically finite, there need not be a specific *number* of slots or shafts or gears. In other words, the *relationships* hold true, regardless of their value. That is the important result. ■

2.3. Application of Peirce's notation to affordances

The above results have already been shown by Peirce. They are merely repeated above to lay a foundation for our intended extension of his logic. We note that affordances as relations already can be described by Peirce's notation.

For example, there are many relations conceivable between users and chairs. Users can sit in chairs, users can carry chairs, users can destroy chairs, and so on. We could describe these relationships as affordances, by calling them sitability, carryability, destroyability, and so on, or by writing out the following:

Chairs afford sitting to users.
Chairs afford carrying to users.
Chairs afford destroying to users.

This latter representation is most conducive to translating into Peirce's notation, with some modifications, as follows.

We state that the type of relation we are considering is an affordance. We can then construct a relation using Peirce's notation quite easily, for example, the relation that chairs afford sitting to users. Following Peirce's original notation, the relation is listed first, followed by all of the conjugates, in order. Thus, $\boldsymbol{a}$ is the affordance relation, s is sitting, c is a chair, and u is a user, then

$$\boldsymbol{a}\mathrm{scu}$$

means "relation of affordance of sitting between a chair and a user."

As with Peirce's original notation, it is important to notice that the relations themselves can take on values. This is of the utmost importance for affordances, where one of the most important properties (see Section 1) is that of *quality*. If we are to be able to perform any meaningful computation involving affordances, we must be able to capture their quality, not only for the sake of documentation, but also so that quality

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

can be optimized. Note that in the case of a negative affordance, the value ascribed to the affordance could simply be a negative number. (The topic of how to determine such values in general is unfortunately beyond the scope of this paper and will be left for future work.)

In other words, it is not sufficient just to be able to determine the existence of an affordance in a Boolean sense. We need the extended logic of Peirce to deal with the relational aspect of affordances.

Thus, depending on the properties of the chair in question, and the properties of the specific user, and the specific mode of sitting (there are many ways to sit in a given chair), then the value of the affordance relation $\boldsymbol{a}$ can be used to capture *how well* that chair affords sitting in that way to that particular user.[2] Note that the value need not necessarily be a number; for example, the "value" could be qualitative, or even emotional.

Generalizing a bit more, if C is a set of chairs and U is a set of users, then

$$\boldsymbol{a}\mathrm{sCU}$$

means "relation of affordance of sitting between chairs and users."

Once again, affordance relation $\boldsymbol{a}$ can be used to capture how well the set of chairs affords sitting to the set of users, but now affordance relation $\boldsymbol{a}$ need not be a single value; it could be up to a matrix of size $\mathrm{C} \times \mathrm{U}$, with a separate value for every combination of chairs and users.

Generalizing even further, if S is a set of various modes of sitting, then

$$\boldsymbol{a}\mathrm{SCU}$$

means "relation of affordance of various modes of sitting between chairs and users."

In this case, affordance relation $\boldsymbol{a}$ could have an additional dimension, in other words, a third-order tensor of size $\mathrm{C} \times \mathrm{S} \times \mathrm{U}$.

REMARK. It should be obvious that $\boldsymbol{a}$csu is a subset of $\boldsymbol{a}$CSU. ■

Turning now to involution, recall that x^{a} means "everything that is an x for every individual of a." Performing involution on the terms in the previous example, we can construct

$$\boldsymbol{a}^{\mathrm{cu}}$$

as meaning "everything that is a relation of an affordance between a chair and a user," in other words, the set of all affordances involving one particular chair and one particular user.

REMARK. We need not know what all of the affordances are in order to manipulate this relation. ■

[2] Strictly speaking, the relation and its value are two different things. Thus, an affordance relation with a value could be given a slightly different notation, such as a', which would have an extra parameter to capture the value explicitly.

Say, for example, that a particular wooden chair affords sitting (s) and carrying (c) and destroying (d), among other affordances, to a particular user, then

$$\boldsymbol{a}^{\mathrm{cu}} = \{\mathrm{s}, \mathrm{c}, \mathrm{d}, \ldots\}.$$

In theory, such a set could be infinite. In practice, the size of the set need not be known. As above, the quality of each affordance could be stored as a value for every member of the set.

REMARK. Note that $\boldsymbol{a}^{\mathrm{c}}$ and $\boldsymbol{a}^{\mathrm{u}}$ are both meaningless, because the affordance relation as defined above requires (at least) two conjugates. Recall from Section 1 that affordances are not properties belonging to a single system. ■

Eleanor Gibson and other researchers (see, e.g., Gibson, 2000) have explored how infants' experiential learning in situated environments can be understood as a process of discovering affordances that can then be perceived later in life. In that framework, learning is simply a process of adding affordances to the set $\boldsymbol{a}^{\mathrm{Xu}}$, where X is the set of objects in the environment with which the infant interacts.

3. AN APPROACH TO COMPUTING AFFORDANCES BASED ON CONTEMPORARY SET THEORY

To discuss the computability of affordances as relations in contemporary terms, we require a short background on what it means to be a relation and what it means to be computable. Fortunately, contemporary mathematics, particularly set theory, provides us with precise definitions of these concepts. The following definitions are based on those discussed by Hedman (2004).

A *set* is defined as an unstructured collection of elements. These elements could be numbers, variables, or anything else, but by definition, the order and repetition of elements within a set do not matter. A set is essentially a database with no structure.

A *relation* on a set is a subset involving elements, in which some meaning (the relationship) is ascribed between the elements of that subset. The number of elements involved determines the "*arity*" of a relation. A one-element relation, say, an adjective pertaining to a single element, is a unary relation. A relation between two elements, say, that one element is larger than another, is a binary relation. A relation between three elements, say, the three strands involved in a braid, is a ternary relation. In general, relations involving n elements are n-ary. Aside from its arity, the definition of a relation (what it means to be braided, for example) is external to the set.

Simple arithmetic operators including multiplication and addition are relations. To capture the output of such an arithmetic operation, however, requires the concept of a function.

A *function* is essentially a mapping from one set to another set. Given any input from the first set (the function's domain), the function will produce at most one output that belongs to

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

the second set (the function's range). A function is typically defined using a combination of relations, such as multiplication, addition, and exponentiation, as well as other functions, such as the trigonometric functions. The concept of arity also applies to a function, in particular to the number of inputs the function takes.

Already it should be apparent from these elementary concepts in set theory that relations and functions are qualitatively different concepts. They are distinct yet complementary.

Turning to the issue of computability, a function is considered **computable** if there exists a computer program of finite steps that can produce the correct output in the range of that function for any input in the domain of that function. It follows that only a finite number of functions are computable because the number of steps available to compute them is limited. Therefore, most functions are not computable. For example, we could define a function love (male, female), which returns true if the individual male and female are in love, and false otherwise. That function is not computable. (One could ask the male and female, but they could lie or give inconsistent answers.)

Fortunately, the kinds of functions commonly encountered in trigonometry, calculus, and so on, are computable. The proof of the computability of polynomials, for example, is commonly offered. It is worth noting that *in order for a function to be computable, the relations used in the definition of the function must also be computable*. Proofs for the computability of the arithmetic relations ($+$, $-$, $*$, $/$) and ordinal relations ($<$, $>$, etc.) are commonly offered, typically involving the concept of recursion.

This brings us to our central concern, the computability of relations. We can approach the computability of relations by involving a function. The **characteristic function** of a relation takes as inputs all the elements used in the relation and returns true if the relation holds, and false otherwise. For example, if the relation is "braided," and three strands are braided together, the characteristic function for braiding would return true. If three other strands are not braided, the characteristic function for braiding would return false. If the characteristic function is computable, then the relation is said to be **decidable**.

Thus, for example, if we can write an algorithm to determine if an affordance relation ***a*** exists, then its characteristic function is computable and we can therefore decide if that affordance exists. By extension, *if we can write an algorithm to determine the value of* **a** *(the quality of the affordance), then I assert that the affordance is computable.*

To me it is an open question as to whether the affordances in which we are interested in design are computable. It follows from the above considerations that only finite affordances are computable. I suspect that design would be more easily automated if certain affordances were computable. The fact that only certain aspects of design have heretofore been successfully automated suggests to me that only certain affordances important to design are computable, while many more are not computable.

This brings us to our final consideration in this section, the concept of what it means to be *intuitively computable*, which basically asserts that if a person can compute a function, then a computer should be able to do likewise using an appropriate algorithm. *Church's thesis* is one expression of this idea, where the algorithms are recursive in nature. *Turing's thesis* is another expression of this idea, where the algorithms are executed in linear order.

I believe there is a strong connection between Gibson's original idea of direct perception derived from psychology and the idea of intuitive computability derived from theoretical computer science. A possible bridge between these ideas is what I would call *naturally computable*. What I mean by naturally computable is a situation in nature that we can observe as happening, but we cannot predict in advance (i.e., compute) using an algorithm. Examples of this are everywhere in nature, from complex systems such as the weather to much simpler systems of entanglement, such as the "*n*-body problem" of determining the paths of *n* mutually gravitationally attracting bodies. Even the path of a single electron has thus far only been described probabilistically, not definitively. Nature has no problem determining any of these situations in real time. Similarly, our minds seem to have no problem perceiving affordances in real time. Hence, direct perception is naturally computable.

Because it is the human mind at work in what is intuitively computable, then by the Church–Turing thesis, direct perception should be computable in the algorithmic sense. These considerations, while informal, support the basic premise of the computability of affordances. In other words, *if humans can perceive an affordance, an algorithm should exist to do likewise, that is, to compute the characteristic function of that affordance and determine its existence and value*. Because the Church–Turing thesis remains unproven, it follows that my application of it to the computability of affordances as relations is likewise unproven, but it can guide our attempts at computing affordances just as the Church–Turing thesis has guided investigations into computer science for decades.

Thus far I have proposed standard set theory as a mathematical tool to aid in the computation of affordances. Chemero and Turvey (2007) have already gone a step further, by suggesting that hyperset theory is necessary to adequately model the affordance relation, because affordances can be self-referential. I believe Chemero and Turvey's approach is deeply compatible with my own, and agree that hyperset theory allows for even richer computations than standard set theory.

4. THE DECOMPOSITION OF AFFORDANCES

In the literature, affordances are often treated individually, and less commonly in groups. In previous work, my colleagues and I have suggested a hierarchical decomposition of affordances is possible, by showing, for example, that the affordance of sustainability could be decomposed into several more specific affordances, such as recyclability and reusability (Maier & Fadel, 2009*b*). In another paper, my colleagues and I discussed in

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

more detail how affordances map to a hierarchical decomposition of a product's architecture (assemblies, subassemblies, parts, etc.; Maier, Mocko, et al., 2009), but we did not investigate the converse, how an artifact and a user map to a hierarchical decomposition of affordances.

I believe Gaver (1991) was the first to use the term *nested affordances*, which he defined as "affordances that are grouped in space." These groups, however, like birds in a nest, were not necessarily in any hierarchy, although Gaver also discusses affordances that are acted upon sequentially in time. Later, McGrenere and Ho (2000) also used the term nested affordances and went further by asserting plainly that "[i]t is important to note that affordances exist (or are nested) in a hierarchy and that the levels of the hierarchy may or may not map to system functions. In other words, affordances do not necessarily map one-to-one onto system functions."

A common limitation in the literature has been the absence of appropriate mathematics to describe the nature of affordances, in this case, the hierarchical nature of affordances. Set theory, as described in the previous section, provides a basic level of mathematics that can be useful. For example, we could define a superset that contains all the known affordances of a product with respect to a certain user. Next we could define sets that contain one or more of those affordances. Each of those sets may overlap. The affordance of recyclability would belong to the set of sustainability, which would also contain the affordance of reusability. However, reusability could also belong to the set of legal requirements.

Design methods as well need to take into account the hierarchical, not just sequential, nature of affordances. This area obviously has a lot in common with decomposition of requirements generally, and may benefit from methods of functional decomposition as well. Determining the appropriate level of decomposition of affordances remains as future work.

In the next sections, we turn our attention to the computability of a specific relation and a specific affordance.

5. REMARKS ON THE COMPUTATION OF PROXIMITY

Most modern artifacts are composed of more than one part. Of utmost importance in the design of such artifacts is the ability to describe the spatial relationships between the various parts. Modern computer-aided design software that implements solid modeling greatly facilitates this process. However, a general limitation of such software is that it captures information about the geometry of the parts but not about their meaning. The ability to capture knowledge about the affordances of various geometries with respect to certain users would be a significant advancement to the field of computer-aided design. Other authors in this issue, namely, Kim and Fadel, Mata, and Mocko, present related work on this topic.

A very simple relation that describes the relationship between two geometries is their proximity. In this section, the computability of the proximity relation is explored, as a very simple example of how a relation can be computed. The proximity relation between points has interesting implications for two-dimensional image processing, which is the focus of this section. The proximity relation and other relations between faces and solids are left to future work.

Given two points a and b, their proximity ***p*** is a relation that belongs to neither point individually. It is complementary, that is, an emergent property of the two point system. Using Peirce's notation, we could write

$$\boldsymbol{p}\mathrm{ab}$$

that would mean "relation of proximity between points a and b."

The value of proximity ***p*** could be used to represent how close together the two points are. This value has limits. Proximity approaches infinity as the two points approach each other, and proximity approaches zero as the two points become increasingly distant.

Proximity also generalizes easily to more than two points; say we add a third point c. Then we could write separately

$$\boldsymbol{p}\mathrm{ab}$$
$$\boldsymbol{p}\mathrm{bc}$$
$$\boldsymbol{p}\mathrm{ac}$$

to describe the proximity between each pair of points, where it is obvious that the proximity relations are noncommutative, that is, knowing the proximity between a and b and between b and c does not determine the proximity between a and c (see Fig. 1).

The proximity of the triplet a, b, c is of the most interest:

$$\boldsymbol{p}\mathrm{abc}$$

which is not summative; that is, it is not determined simply by adding ***p***ab and ***p***bc and ***p***ac. Referring back to Figure 1, it is also obvious that the image on the left has a different proximity ***p*** value than the image on the right. Not only are the points on the right closer together on average than those on the left; the arrangement of the points is qualitatively different in each case.

Thus far we have avoided a direct measurement of proximity, even though we could imagine a function involving, for example, the inverse of distance. It is worth noting that no scale is provided in Figure 1. The proximity relation is thus in some sense scale free. In other words, the proximity of the points, a

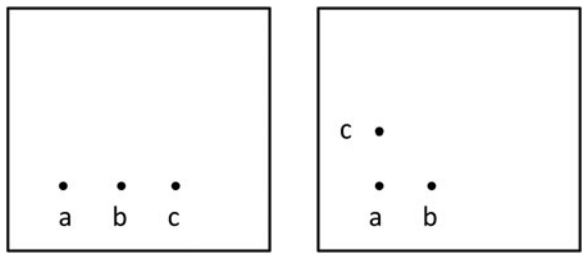


Fig. 1. Noncommutativity of the proximity relation.

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

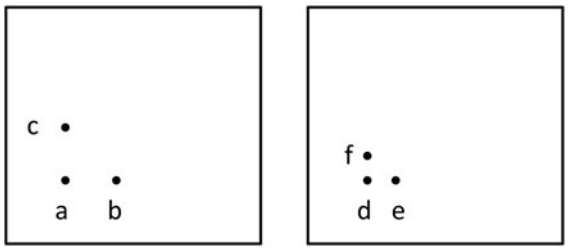


Fig. 2. Ordinal scale of the proximity relation.

property of the multipoint system, is independent of the distance from which the multipoint system is observed.

Lacking a cardinal function (as is often the case with affordances, where no definitive measurement is known), an ordinal ranking could still be achieved, for example, of the two images in Figure 2.

We could assert that the points in the image on the right are in closer proximity than the points in the image on the left, and hence

$$\boldsymbol{p}\text{abc} < \boldsymbol{p}\text{def}$$

and as more than two such relations are compared, certain statistics become available, such as median and mode, but not mean.

These properties of proximity scale up as the number of points increases, up to arbitrary many points, such as a binary image, or a three-dimensional nodal structure. This line of reasoning has been applied to analyze the presence and quality of relations in binary (black-and-white) images, but that is somewhat beyond the scope of this paper. In the next section, we present a simple case of computing an affordance.

6. COMPUTATION OF AN ARTIFACT–ARTIFACT AFFORDANCE

6.1. Background

In earlier work my colleagues and I (Maier & Fadel, 2007) have discussed the trouble with affordances, which is not that they are ill defined, but that they are so very broad. What affordances exist between two interacting systems can be just about anything, depending upon the two systems involved. The key to understanding what is actually afforded in a given situation is form dependence; the *structure* of the two systems (internally and externally) determines what affordances exist, and their respective quality. This key unlocks the concern raised by Brown and Blessing (2005) early on about the affordance-based design approach, that "while affordances, as 'possible actions,' are an important consideration while designing, it isn't always easy to reason out what they are, as the search space is large." In other words, the reasoning (whether done by a human designer or facilitated by a computer) must be based on the structures of the two interacting systems (say an artifact and a user). The structure narrows the search space sufficiently such that human designers do not become lost. (See Shu et al. in this issue for several methods by which affordances can be successfully identified.) The same should be true of a computational system. Therefore, let us consider two simple artifacts, and attempt to demonstrate the characteristic function for an affordance relation between the two artifacts.

6.2. Affordance of turnability of gears

Let us say we are interested in whether or not two gears afford turning. This is an example of an artifact–artifact affordance rather than an artifact–user affordance just because the structure of an inanimate gear is simpler to describe for illustration purposes than the structure of a human user.

Engineering science already dictates necessary conditions for two gears to transmit power effectively. For two spur gears, they must be placed in the same plane; they must have compatible teeth profiles; and their spacing must be such that their two pitch circles are tangential (see Fig. 3). Thus, the algorithm for checking the characteristic function for gear turnability merely needs to check these three conditions. This would be easy enough to implement in a conventional computer-assisted design solid modeling environment. (It is interesting that the

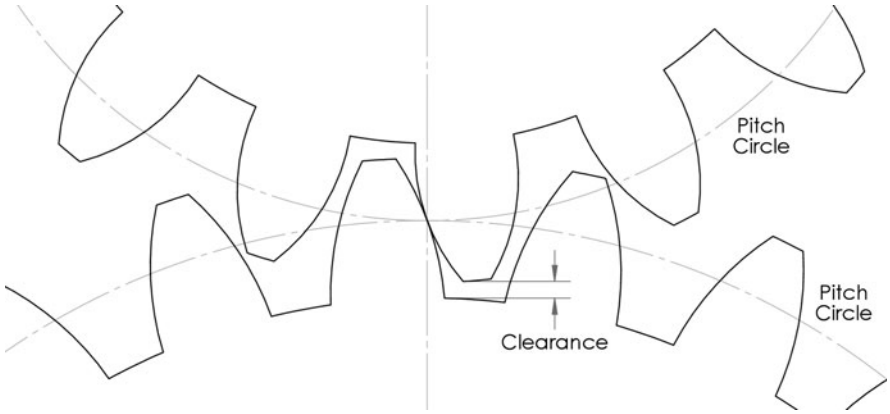


Fig. 3. Gear spacing.

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press

software I use and teach is not yet this smart. It will happily animate gears turning that do not touch at all, or severely overlap.)

The third condition above (that the spacing between two spur gears must be such that their pitch circles are tangential) could be relaxed somewhat. If the two pitch circles overlap slightly, but there is still nonzero clearance, then the two spur gears still afford turning, but not as well as two gears with tangential pitch circles. There will be more friction than necessary, and hence less efficient power transmission. Given two gears a and b, their turnability $\boldsymbol{t}$ is greater (and optimal in this simple scenario) if their pitch circles are tangential. Given two gears c and d, whose pitch circles overlap somewhat but retain some positive clearance, then

$$\boldsymbol{t}\mathrm{cd} < \boldsymbol{t}\mathrm{ab}.$$

The values of $\boldsymbol{t}$cd and $\boldsymbol{t}$ab could be determined experimentally by measuring their efficiency at power transmission, or predicted theoretically using a friction theory, or computationally derived using modeling software.

7. CONCLUSION

Several approaches to computing affordances have been discussed in this paper. Using Peirce's notation, affordances as relations may be manipulated mathematically. For example, specific affordances could be combined into an objective function that is subject to optimization. Peirce's notation is consistent with contemporary set notation, which leads into considerations of formal computability. Based on the formulation of an affordance as a relation in a set-theoretic sense, I asserted that if an algorithm can be written to determine the quality of an affordance, then that affordance is computable in theory. Then based on a novel definition of what it means to be naturally computable, and the Church–Turing thesis, I have proposed that if humans can perceive an affordance, an algorithm should exist to do likewise, that is, to compute the quality of an affordance. Two applications of computing relations have been offered: the proximity relation and the affordance of gear turnability. Many possible avenues for future work have also been suggested, hopefully providing direction for future development of affordance-based design.

REFERENCES

Brooks, R.A. (1991). Intelligence without representation. *Artificial Intelligence 47(1–3)*, 139–159.

Brown, D., & Blessing, L. (2005). The relationships between function and affordance. *Proc. ASME 2005 Int. Design Engineering Technical Conf. & Computers and Information in Engineering Conf.*, Paper No. DETC2005-85017, Long Beach, CA, September 24–28.

Chemero, A., & Turvey, M.T. (2007). Gibsonian affordances for roboticists. *Adaptive Behavior 15(4)*, 473–480.

Gaver, W.W. (1991). Technology affordances. *Proc. CHI'91*. New Orleans, LA: ACM.

Gibson, E.J. (2000). Perceptual learning in development: some basic concepts. *Ecological Psychology 12(4)*, 295–302.

Gibson, J.J. (1979). The theory of affordances. In *The Ecological Approach to Visual Perception*. Hopewell, NJ: Houghton Mifflin.

Hedman, S. (2004). *A First Course in Logic: An Introduction to Model Theory, Proof Theory, Computability, and Complexity*. New York: Oxford University Press.

Jones, K.S. (2003). What is an affordance? *Ecological Psychology 15(2)*, 107–114.

Maier, J.R.A. (2006). How structure influences behavior. *General Systems Bulletin 35*. International Society for the Systems Sciences.

Maier, J.R.A., Anandan, S., Bapat, V., Summers, J.D., & Bettig, B. (2007). A computational framework for semantically rich design problems based on the theory of affordances and exemplar technology. *Proc. Int. Conf. Engineering Design, ICED'07*. Paris: Design Society.

Maier, J.R.A., Ezhilan, T., & Fadel, G.M. (2007). The affordance structure matrix—a concept exploration and attention directing tool for affordance based design. *Proc. ASME 2007 Int. Design Engineering Technical Conf. & Computers and Information in Engineering Conf.*, Paper No. DETC2007-34526, Las Vegas, NV, September 4–7.

Maier, J.R.A., & Fadel, G.M. (2007) Identifying affordances. *Proc. Int. Conf. Engineering Design, ICED'07*. Paris: Design Society.

Maier, J.R.A., & Fadel, G.M. (2009*a*). Affordance based design: a relational theory for design. *Research in Engineering Design 20(1)*, 13–27.

Maier, J.R.A., & Fadel, G.M. (2009*b*). Affordance based design methods for innovative design, redesign and reverse engineering. *Research in Engineering Design 20(4)*, 225–239.

Maier, J.R.A., Mocko, G., & Fadel, G.M. (2009). Hierarchical affordance modeling. *Proc. Int. Conf. Engineering Design, ICED'09*. Stanford, CA: Design Society.

McGrenere, J., & Ho, W. (2000). Affordances: clarifying and evolving a concept. *Proc. Graphics Interface 2000*, Montreal, May 13–17.

Murphy, R.R. (1999). Case studies of applying Gibson's ecological approach to mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 29(1)*, 105–111.

Peirce, C.S. (1873). Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic. *Memoirs of the American Academy of Arts and Sciences, New Series 9(2)*, 317–378.

Rome, E., Hertzberg, J., & Dorffner, G. (Eds.) (2006). *Towards Affordance-Based Robot Control*. Berlin: Springer–Verlag.

Stoffregen, T.A. (2003). Affordances as properties of the animal–environment system. *Ecological Psychology 15(2)*, 115–134.

Turvey, M.T. (1992). Affordances and prospective control: an outline of the ontology. *Ecological Psychology 4(3)*, 173–187.

Warren, W.H. (1984). Perceiving affordances: visual guidance of stair climbing. *Journal of Experimental Psychology: Human Perception and Performance 10(5)*, 683–703.

Jonathan R.A. Maier is a Lecturer in the General Engineering Program at Clemson University. Dr. Maier's broad primary research area is design theory and methodology, and affordance-based design in particular, although he also enjoys a wide variety of design-related activities, including product development and the history of designed artifacts.

https://doi.org/10.1017/S0890060415000207 Published online by Cambridge University Press