Can an Algorithm be Unethical?

Christian Sandvig*, Kevin Hamilton†, Karrie Karahalios†, Cedric Langbort†

*Department of Communication Studies and School of Information, University of Michigan
†Center for People & Infrastructures, University of Illinois at Urbana-Champaign

Paper presented to the 65th annual meeting of the International Communication Association, San Juan, Puerto Rico, USA. May 2015.

Corresponding author: Christian Sandvig
University of Michigan
Room 5385 North Quad
105 South State Street
Ann Arbor, MI 48109-1285
USA

Tel.: +1 (734) 763-0861
FAX: +1 (734) 764-3288
e-mail: csandvig@umich.edu

Can an Algorithm be Unethical?

Abstract

In Information and Communication Technologies (ICTs), computer algorithms now control the display of content across a wide range of industries and applications, from search results to social media (Gillespie, 2013). Abuses of power by Internet platforms have led to calls for "algorithm transparency" and regulation (Pasquale, 2014). This paper responds by asking what an analyst needs to know *about algorithms* in order to determine if an ICT is acting improperly, unethically or illegally. It further asks whether "the algorithm" is a useful object for ethical investigation. The paper briefly reviews the technical history of the term, then performs an ethical analysis of a hypothetical surveillance system to investigate the question of whether it is useful to consider an algorithm "unethical" in itself. It finds that law and policy researchers can in fact employ technical expertise about algorithms and that such expertise might be crucial to make judgments about future ICTs.

Keywords:

applied ethics

Information and Communication Technologies (ICTs)

science and technology studies (STS)

Internet studies

Can an Algorithm be Unethical?

Hewlett-Packard (HP) suffered a serious public relations crisis when it was revealed that their implementation of what was probably a *bottom-up feature-based face localization algorithm* (Yang et al. 2002) did not detect black people as having a face (Simon 2009). Cameras on new HP computers did not track the faces of black subjects in some common lighting conditions; conditions that were easy to reproduce. In an amusing YouTube video that as of this writing has 3 million views, Wanda Zamen (who is white) and Desi Cryer (who is black) demonstrate that the HP camera eerily tracks Wanda's face while ignoring Desi, leading Desi to exclaim jokingly, "Hewlett-Packard computers are racist."[1]

Dyer (1997), a historian of cinema, famously came to a similar conclusion about photographic technology. In his classic study of photographic and cinematic representation of human skin, he explains that as early photographers turned to portraiture in the 1840s, "[e]xperiment with, for instance, the chemistry of photographic stock, aperture size, length of development, and artificial light all proceeded on the assumption that what had to be got right was the look of the white face" (ibid.). Technological investments followed value: to put it starkly, black people weren't considered to be worth photographing. At the dawn of the camera the representation of white skin was seen as very difficult due to its tendency to "wash out" or shade to unrealistic red tones, but with effort these difficulties were solved, white became the norm, and photographing non-white people has been considered the problem or the exception ever since. As late as the 1970s, 3M Corporation developed a television signal named "skin" (a pale orange color) used to grade the quality of videotape (ibid.).

These two cases are obviously different—Wanda and Desi demonstrated an embarrassing HP oversight that was quickly corrected with a software update. In contrast, Dyer unearthed evidence that decisions made in the early days of photography embedded racist assumptions in chemical

[1] http://www.youtube.com/watch?v=t4DT3tQqgRM

formulations, apparatus, and process that led to 150 years of difficulty for non-white photographic subjects (and counting). Yet their similarities are also instructive. In each instance a technological system becomes the vehicle for embedded human social dynamics (racism), which it could then perpetuate. In both cases the system produces effects that are clearly ethically worrisome. Considering intent or assigning human responsibility is probably equally fruitless in both cases.[2] Understanding the dynamics of both cases in details requires some engagement with the technical innards of the system.

Yet Dyer manages his analysis and critique without resorting to the word or the idea "algorithm."[3] Today, computer algorithms play a critical role in producing and curating our communications and shared culture (Barocas et al., 2013). They determine how our questions are answered (Introna & Nissenbaum, 2000); decide what is relevant for us to see (Gillespie 2012); craft our personal and professional networks (Hamilton et al., 2014); suggest who we should date and what we should watch; and profile our behavior to determine what advertising we will receive (Bermejo 2007). Research and writing is now blossoming among academics, journalists, and non-profit organizations that criticizes (Pasquale, 2014; Gangadharan, 2014) and even reverse engineers (Hannack et al., 2013; Angwin, 2013; Diakopoulos, 2014) these algorithms in an attempt to see inside these new systems and understand their consequences (for an overview, see Sandvig et al., 2014). Law and policy scholarship has also raised the question of whether or not governments now need to compel ICT providers to reveal algorithms -- a movement known as "algorithm transparency" (Pasquale, 2014). There is some precedent for administrative law regulations that forbid particular processes in algorithms: in the history of aviation, travel booking systems have been forbidden to deploy sorting algorithms that display more profitable itineraries over shorter, cheaper itineraries (Sandvig et al.,

[2] That is, neither 3M nor HP as organizations nor the individuals within them probably intended to "be racist." If we identify specific inventors of photographic apparatus from the 1840s they would only be exemplars of the attitudes common at that time. (The complete lack of black skin in whatever HP department develops webcams does seem to be an impressive error, however.)

[3] Whether photography in the 1840s had algorithms or not is a question that will be addressed later in this paper.

2014). In sum, policy scholars are now arguing for the regulation of algorithms as a distinct source of harms (Grimmelmann, 2009). While there is a well-developed field of information ethics, it does not consider algorithms *per se* (Floridi, 1999).[4] The rise of this research topic also raises an important general question: How much do researchers who study these systems actually need to know about algorithms? That is, is the current use of the word "algorithm" a trendy buzzword, a signal that there is a new formation of our media systems writ large, or does the word actually indicate that the careful analyst should understand and think about specific instances of computer algorithms as a computer programmer might understand the term?

In the past, cultural critics have studied selection decisions made in the media industries without, for example, reverse engineering the neurobiology of the brains of television executives or newspaper editors. Although "the algorithm" has emerged as an important concept in the public mind (Striphas, 2014; Sandvig 2014), it also seems reasonable that scholars of "algorithmic culture" (a phrase coined by Galloway, 2006) might study the consequences of the addition of computing to these media and information systems by studying the socio-technical systems at a macro-level. In contrast, we argue in this paper that there is important new knowledge to be gained by considering the operation of algorithms writ small--the technical details of the innards of particular computer code. To accomplish this analysis we have assembled a multi-disciplinary team of co-authors (a computer scientist, an applied mathematician, an artist, and a social scientist) who will discuss the problems of assigning ethical responsibility to algorithms in general, then review one case in detail--a hypothetical surveillance system. In writing this analysis we sought to answer the question: "Is it useful to ask if 'an algorithm' is unethical?" and "Is 'the algorithm' a useful focus of legal or ethical scrutiny?" As we shall see, we find the answer to both questions is yes.

[4] Conceptually an "algorithmic ethics" might simply be another lens into information ethics, as that field deals primarily with data as its object, while we presume an algorithmic ethics would deal with processes that act on data. Nonetheless, the current boom of scholarship on "algorithms" presses us to consider whether or not this word or object "the algorithm" deserves special analysis and that is the subject of this paper.

Returning to the situation of HP's "racist" digital camera, the manufacturer had a wide range of pre-existing approaches to *localization* at its disposal (Yang et al. 2002), localization being the term for the problem of detecting the presence and location of particular features (faces) in an image, in contrast to *recognition* (matching them to another face). HP could have chosen a *bottom-up feature-based* algorithm—likely the intellectual descendant of Chetverikov & Lerch (1993), who realized that human faces could be successfully detected by asking a computer to find a pattern of dark and light "blobs," indicating the eyes, cheekbones, and nose. As this tactic depends on contrast between the skin and the whites of the eyes, dark skin presents a challenge.

Other algorithms for locating a face represent different trade-offs. Detecting faces via a predefined palette of skin colors is an alternative approach (Graf et al. 1995) – and using *histogram intersection* between a control "skin" and a target will obviously be very sensitive to the definition of "skin color" from the outset. Since these approaches require a definition of what counts as "skin" in advance, it logically follows they could certainly be racist. In contrast, *edge detection* methods might be much more dependent on the contrast between the body and the background. They could be sensitive to hair color and a contrasting background color, making skin color relatively unimportant. A third approach would be an algorithm for determining and displaying skin color based on machine-learning. This might seem a desirable solution, but we can just as easily imagine such a process coming to objectionable conclusions on its own depending on the training data it was given.

While each of these algorithms has consequences for efficiency and accuracy when evaluated in specific situations, we can also say that the choice of algorithm delegates the ability to define "what is a face?" and that this question is ethically salient. Though one could argue that an ethical critique of algorithms should focus more on the design process of a system than on the resulting technology, we believe that the nature of algorithms invite a closer examination of that very distinction. To neglect the algorithm itself in the study of a system's ethics overlooks the significance of algorithms as singular

"statements[5]," ways of framing problems and assuming not only particular implementations, but ways of dealing with undesirable results. As such the algorithm itself offers unique opportunities for understanding the relationship of theory (conventionally regarded as the domain of *algorithms*) and action (conventionally, the domain of *programs*). In the rest of this paper we will try to elaborate the position that algorithms are an important point of ethical intervention information and communication technologies (ICTs), and that investigating them might require a specialized approach.

Defining Algorithms

In 1966, a very brief letter to the editor appeared in the *Communications of the ACM*, a leading periodical in the nascent area of academic computer science (Wangness & Franlin 1966) :

> "We are making this communication intentionally short to leave as much room as possible for the answers.
>
> 1. Please define 'Algorithm.'
> 2. Please define 'Formula.'
> 3. Please state the difference.
>
> Signed: T.Wangness & J. Franklin."

Two answers were provided. The first, by Hartmut Huber, opens as follows:

> "Before the concept 'algorithm' was defined precisely, it was understood in an intuitive sense as a finite sequence of rules operating on some input yielding some output after a finite number of steps. Since Turing, Kleene, Markov and others, we have several precise definitions which have proved to be equivalent. In each case a distinguished sufficiently powerful algorithmic language (or programming language) is specified and an algorithm is defined to be any program written in

[5] Andrew Goffey on the algorithm as *statement*: "For Foucault, the statement is not analytically reducible to the syntactic or semantic features of a language; it refers instead to its historical existence and the way that this historical existence accomplishes particular actions." (Fuller 2008)

this language (Turing Machines, mu-recursive functions, normal Markov algorithms, and so on) terminating when executed."

The second answer, offered by Donald Knuth (who went on to write the monumental and widely-read book *The Art of Computer Programming*), states:

"To me the word algorithm denotes an abstract method for computing some function, while a program is an embodiment of a computational method in some programming language."

Several points are worth taking away from the introspection of these computer scientists, trying to define the very subject of their science. First, no mention of "a computer" (or "hardware" or "equipment") is made at any point. It appears quite clearly that, to a computer scientist, algorithms exist independently from any kind of physical substrate on which they may be implemented. Even in the original framing of Wangness & Franklin, disambiguation is sought not between "what a computer does" and "what other mechanisms do" but between "algorithm" and "formula," two disembodied notions.

Knuth's answer (and, to some extent, the original question) draws two further distinctions: between an algorithm and the function it performs, and between an algorithm and the program that implements it. The latter one is of the same nature as the point above in that it separates the algorithm from its embodiment, be it hardware, software, or code. Note that the same algorithm can be implemented differently in two different programming languages.

One can see here the tensions at work between the algorithm as an "abstract" set of rules or a strategy (Knuth) and the algorithm as a manifestation of such rules (Huber) in a particular "algorithmic language." Between these two answers, there exists at least one level of abstraction between a computer program and its algorithmic origins, and possibly a second layer of abstraction, between a problem or "function" and its expression in an algorithmic language suitable for bringing to a particular computational process. (One remembers here, in early computing, that computing labor itself was

divided between the mathematicians who devised a problem, and the programmers who set up the computer to address the problem.)

We will later make the case that the study of an algorithm's ethics will require examination of both its composition and its consequences. Yet the early discussion of the concept of the algorithm cited above shows that, even for a seasoned computer scientist, it can be tricky to even know when you are talking about the algorithm, its implementation, is effects, or its function.

Being specific in our questions about ethics

By taking the computer scientist's view of what an algorithm is and is not, it is thus apparent that the question "is an algorithm ethical?" is a very focused one because "the algorithm" is a term that is narrowly defined in computer science. It asks specifically about a certain kind of *process* or *strategy* and not about the goal itself. This isn't to say that the goal isn't important, clearly it is critical, yet asking about the ethics of the goal is simply a different question than considering the ethics of an algorithm. Asking "is this surveillance algorithm ethical?" is a different kind of question than "is surveillance ethical?" and one question can't be substituted for the other.

We need to evaluate the algorithm by its properties and processes--and this requires attention to some of the specific features and characteristics of algorithms. We need to look at an algorithm's particular inputs and potential outputs, and yes, even its ethical relationship to "good." To understand an algorithm as "oriented to good" is to trace possible effects back to causes - the actual processes at hand, which though not reducible to effects, do reveal an algorithm's teleology and potentially its deontological duties (as we will discuss later). And though we will refrain from asserting "mindfulness" to algorithms and we do not endorse a deterministic view of technological effects, we will show how an algorithm might be structured in a way that tends toward or against normatively positive behavior.

Case Study : Race and surveillance algorithms

To make this discussion more concrete, consider the problem of evaluating a surveillance algorithm as ethical in its treatment of race. In our introductory example, photographic apparatus was found to contain embedded ideas about representing race probably without the knowledge of the designers. Moving now to a different kind of consideration of race, take the example of a surveillance system designed to evaluate streams of surveillance camera footage in airports and train stations and decide under certain circumstances to notify a human operator that something worthy of notice may be happening.

Race is obviously fraught in this context of surveillance, public safety, and counterterrorism. While race has an accepted as a part of the visual description of people (e.g., on identity documents),[6] even the presence *of the idea of race* can be seen as prejudicial in the context of, say, a screening process at an airport or train station that is not searching for a particular person. This makes race a useful test case to investigate the question of whether or not surveillance algorithms can be judged ethical.

Using accepted terminology often used to discuss algorithms, we will distinguish between algorithms by defining *inputs* and *outputs*. We understand the *inputs* in this case to be the criteria by which the algorithm selects events for possible consideration and processing, by the human operator or a computational process. We understand the *outputs* in this case to be the images isolated from surveillance feeds and displayed to the operator for consideration and judgment.

We understand *racism* in this case to be an inappropriate or unmerited consideration of race as

[6] There is a large literature on the problem of race as an indexical descriptor of someone's appearance. However, we will not treat this as an ethical problem here as it is likely to be accepted in the airport and train security community that connecting people's skin color to an idea called "race" is a necessary way to facilitate their visual identification. (To consider the classification of race on identity documents at all as a problem, see Bowker & Star [1999] on South Africa under apartheid [ch. 6]).

a factor in determining possible threats to public safety, as determined either by looking at the consequences of a particular algorithm or its rules. We will imagine that the aims of the system are to avoid racist means *or* ends, and that the human operator in the system has been trained to disregard race entirely. Based on the framework above, we imagine the following possible scenarios for the algorithm at hand.

Race as an input

Here, we'll imagine a version of the system wherein computer vision algorithms select persons for review by the operator based on skin color, as defined through hue and saturation of pixels within shapes that infer the presence of exposed skin in a human form. For example, one such algorithm might 'normalize' lighting in images, look for "elongated areas representing limbs and connected limbs constrained by a skeletal geometry" and then search for skin by looking for "targets $110 < \text{hue} < 150$ where $20 < \text{saturation} < 60$ and $130 < \text{hue} < 170$ where $30 < \text{saturation} < 130$" (Fleck & Forsyth 1996).

Considering the consequences of this algorithm would lead us to conclude that in some situations, the algorithm could produce racist results.[7] It is quite plausible that this sort of algorithm could, e.g., only reliably target dark-skinned people while not seeing other lighter-skinned people as "people" (or vice versa). Indeed, in the example above by Fleck and Forsyth, their initial formula for estimating skin color was too heavily weighted toward white skin, and so had to be adjusted. Even an accurately targeted algorithm intended to retrieve all persons of caucasian skin color (a dubious possibility) could then be used for racist purposes, resulting in racist outcomes.

But we could not conclude that this would produce racist results in all situations. There are, for example, some situations where race is an acceptable criterion for filtering the occurrences of human forms on a live video feed. For that matter, skin color on a camera charge-coupled device is not even a

[7] That is, a consequentialist analysis.

consistently accurate indication of race. Skin tone effects in image processing algorithms have also been shown to be extremely sensitive to lighting--identical instantiations of this algorithm in Finland vs. Singapore might lead to only one system being racist due to the quality of the natural light.

This means that we can determine that an algorithm's use of race is unethical--it produces inappropriate consideration of race when implemented in a particular system, but this requires consideration of a variety of circumstances external to the algorithm to make this conclusion. The algorithm, by this reasoning, cannot be racist "by itself" as it is impossible to determine what the algorithm is doing without more information about its implementation and context. Not to mention that whether or not consideration of race is "inappropriate" is a judgement that could be said to depend on the application. Nonetheless, if the system as a whole is unethical and racist, at the least we can say that the algorithm "helps" the system achieve this status.[8] The fact that this example algorithm is designed around judgements about skin tone is a technical detail that the ethical analyst would ignore at their peril.

In comparison, a different approach to applied ethics asks us to consider whether there are racist, unethical things that an algorithm might do regardless of their consequences in a particular context.[9] This question is at the heart of race as a legal problem. In the US, the Civil Rights Act of 1964 was seen as a major victory because it decreed that race simply could not be considered in some situations, regardless of the context, nuances, or consequences. The same color-blind logic (race may not be considered) was later used in California and Michigan ballot initiatives in 1996 to dismantle affirmative action in higher education. Of course the proponents of these measures considered the likely consequences, and they judged this rule about process would produce the consequences they desired. But they also each grappled with the argument that it might be an ethical duty to not consider

[8] Apologies to Slack & Wise (2005) who use this term from Eddie Izzard comedy routines to explain the causality of gun violence (the gun "helps").
[9] That is, a deontological analysis.

race at all, as this has proven to be a potent and vexing argument in the ethics of civil rights (Bonilla-Silva 2009).

When writing a surveillance algorithm, a case could be made under this logic for forbidding the consideration of skin tone. A concerned engineer might argue that it is a bad idea to bring more consideration of race into the world. Even if this algorithm could be applied in a scenario where the effects were not racist, this wouldn't matter. Even if this algorithm could be used in situations where racial discrimination was necessary and justified (such as identifying a particular person by using skin color as one of many attributes), this is not relevant.

Although this is not our personal position, a variety of groups have seen it as a moral duty to not consider race when making judgements of any kind. This ethical position is useful to elucidate some kinds of reasoning that may be particularly salient to judgements about algorithms. Since some computer scientists define an algorithm to be about process alone, violating rules about particular processes that are forbidden are a specific way that an algorithm can be judged unethical or illegal. Note that by this ethical position the algorithm is not just unethical, it can potentially be *independently* unethical--if any consideration of race is unethical or illegal, the algorithm can be racist "on its own" without considering an implementation, program, or context.

There is some logic to this even if we bring consequences back into the frame.[10] This algorithm will inevitably select some persons based on a race they do not identify with, given the subjectivity of race as a category. Whenever applied in a situation where race is not an appropriate criterion for selection, this algorithm will always follow a racist rule. Again the inclusion of skin tone in the Fleck & Forsythe localizer is critical to the analysis. For a certain kind of strict ethical thinker, it is the most important fact that can be revealed about the system.

[10] We emphasize that applied ethical reasoning typically employs a mixture of deontological and consequentialist approaches and that most positions can be justified in both ways. We will discuss this in greater detail later. The deontological frame, here, simply helps us foreground the idea that algorithm ethics could be more rigid duty rather than as a set of outcomes.

Race as a learned conclusion

In a machine-learning process, an algorithm might begin to form conclusions based on race, and then act on those conclusions in determining which processes to apply to which events or subjects. Machine learning algorithms are those that use a statistical evaluation of a set of data to develop decision rules, rather than specifying all decision rules in advance. A classifier machine learning algorithm might be designed, for example, to learn from the images an operator spends more time investigating, and then present a greater proportion of such images, based on criteria that the algorithm "discovered" statistically.

If this machine learning algorithm is giving weight to race - *without even having a category for race* - the algorithm has a strong potential of resulting in racist consequences, without starting out with a racist rule. Even though the output is not an explicit race category or variable, in effect the algorithm would be internally recognizing race because it correlates highly with the desired output. If such an algorithm were applied in a surveillance system designed to notify a human operator of suspicious activity, the operator may end up being shown only subjects of a particular race without ever knowing that such a criterion were in play. A training dataset produced by a racist operator would produce a racist machine learning algorithm. Note that this is not necessarily an argument that must be about consequences. Such a classifier algorithm might not start out with any racist rules, but it can acquire them later via machine learning. Once those rules exist, we can say that the algorithms considers race even if the word "race" is not used.. While this might not be foreseeable by the designer of the classifier, the person deploying the classifier could have a duty to consider the training data to ensure that race is not considered, even though it was never named.

Race without a predefined influence on outputs

How a racist algorithm's logic is exposed to the user might change our evaluation of its consequences. We might imagine a third variation where a computer vision algorithm is designed to search for subjects based on race, but the algorithm only saves this information for later analysis by a second algorithm, an internal or external process. Such a system might even present this information for possible use by the operator, pending a particular event or input. For example, in the previously described surveillance system, the algorithm might be designed to first detect an image of an "abandoned parcel" (left luggage or a potential bomb) and then search for the figure last seen near the parcel, based on analysis that incorporates skin color. The algorithm might also look for multiple patterns in a feed and then present them for "tuning" by an operator in the construction of an output feed.[11] In such scenarios, where race is available to an algorithm as an input, but not clearly connected to outputs, we can see potential for racist consequences, but perhaps less potential. A classifier algorithm could, in other words, return images to the surveillance system operator already sorted, or it could return to operator a choice of possible ways to sort the data. The latter would lessen the possibility of racist results.

Learning from potential racism to enact safeguards

[11] This example is related to the Salient Stills methodology, which consists of a sequence of algorithms to (Teodosio & Bender 2005): (1) determine the global camera position from the video (2) build a three-dimensional space/time representations of the video data (3) create a high resolution image representing the full background (4) extract moving object (5) composing the final still with salient objects. The algorithm allows the user to set parameters for prioritizing salient features that appear in the final still. There are several places where the user can guide the algorithm. For example, in (1) if the user knows the type of shot (zoom, pan, etc.) this information could be used to aid in recovering global camera position. In step (3) the user can choose an operator (mean, mode, first pixel, last pixel) to combine the pixels corresponding to the same spatial location from the three-dimensional video representation. A mean operation will result in translucent renderings of moving objects. A median operation typically removes 'ghosts'. Step (4) can further benefit contextual information from the user. User input can help determine which foreground objects should be discarded or kept. A final compositing (5) tells the story of the image sequence in one image. Again, mean, mode, first pixel, and last pixel operators may be used to render the final image. Salient extracted objects can be composited into the still.

Through close study of how the above scenarios might be more or less likely to result in racist outcomes, we might look to design better safeguards into such algorithms. Some precedent exists for such care in algorithm design. In computer security or health systems, for example, sandboxing processes and other safeguards lessen the risk that systems suffer serious security or privacy failures. Though such approaches might begin to take us into system design rather than algorithm design, we can point to some specific cases in the above examples where the addition of a few new inputs, outputs or steps in the algorithm could result in a process which tends to be more ethical. Indeed, without an adequate consideration of the algorithms the system design process itself is impoverished, as selection of the algorithms to be employed is a major task.

Consider a version of the algorithm in the video surveillance scenario that could receive feedback from the operator with respect to false positives or false negatives. These additional labels from the operator then become extra training data for the system. An adaptive system could incorporate this feedback without the need for fully retraining the system. An algorithm might be designed to specifically detect if the operator is racist, giving the system the ability to "learn" from the operator's own racist conclusions to prevent racist outcomes in the future.

On Virtues, Recipes, and Consequences: Reasoning About Ethics

We mean our reasoning about algorithms to be a practical endeavor, as "the algorithm" as an idea has captured the scholarly imagination at this moment because of its actual implementation in real technological systems whose consequences are felt daily. This discussion intends to address critics, scholars, and analysts, but also algorithm designers and system-builders with applied problems. However, practical ethics is an impractically messy place where hypotheticals and abstractions abound. To conclude our analysis from hypotheticals, we need to clarify what we might mean by asking whether or not an algorithm is ethical, as a variety of forms of ethical reasoning should be considered.

This section will begin with some elementary examples. The point here is not to approach ethics abstractly as a philosopher might, but to investigate the kinds of reasoning about ethics that people actually do. For instance, although they may not be aware of it, each member of the the Association for Computing Machinery (by its own account, the world's largest educational and scientific computing society) is required to abide by a code of ethical conduct.[12] The ACM Code, like most efforts in applied ethics, attempts to grapple with moral problems by borrowing tactics from every major branch of the philosophy of ethics.

ACM members are required to abide by three very different kinds of ethical precept. They are asked to "be honest" (1.3), a form of ethical reasoning a philosopher would recognize as *virtue ethics* traceable to Aristotle, grounded in the concepts of character. Although no computing equivalent of Hippocrates has yet emerged, ACM members are required to "avoid harm," (1.2) a form of *teleological ethics* asking computer scientists to reason about ends or consequences. Indeed, they are explicitly asked to "assess...social consequences" (1.2), and this form of ethics is recognizable as *consequentialism*.[13] Finally, ACM members are also asked in very specific terms to respect copyright (1.5), an example of *deontological ethics* grounded in the idea of following a predefined set of moral rules because of those rules.

These are three different kinds of justification--it is important to note that all three handle the same questions of morality as their content. For example, we could abstain from lying because it is a part of our character to be honest (*virtue ethics*), because lying produces some harmful consequence (*consequentialist*) or because there is a norm that forbids it, regardless of the situation (*deontological*). In real-life problems, such as the design of surveillance systems, people typically proceed using a hodge-podge of all three justifications. Although these three kinds of reasoning are logically

[12] See http://www.acm.org/about/code-of-ethics
[13] From here forward we will use *consequentialist* to stand for all *teleological* ethics as it is easier to remember.

incommensurate when taken in their strict sense, in practice they are not interpreted strictly.[14]

Our conclusion in this paper overall is that it is possible and useful to perform an ethical assessment of an algorithm. With respect to the ACM Code this moves an analyst's focus away from the computer scientist and into the computer, or more specifically into the ideas that comprise the actions of computers (as will be explained later). If we consider an algorithm to be an independent agent (Callon, 1986) we find that these three ethical approaches provide varying degrees of analytic traction.

Algorithms With Vice and Virtue: a Difficult Proposition

It might initially seem strange to ask a particular *bottom-up feature-based face localization algorithm* to "be honest" or exhibit any other kind of virtue. However, the computer science jargon is full of references to "fairness" (a routing algorithm may be seen as fair, for example, if it gives router access to different flows in a round-robin fashion, thus giving each one a `fair' shot, or if it aims at maximizing the social welfare of a group of agents) or "greediness" (for an algorithm handling tasks or following paths of higher perceived value first) to name just two terms related to virtue ethics. We have not encountered an algorithm that is characterized as "modest," "arrogant" or "vindictive" but it probably should not be ruled out.

These characterizations are more used as a way to describe some of the basic techniques underlying an algorithm than as bona fide ethical judgements, however, and computer science and virtue ethics part ways when these ideas are considered more than superficially. Virtue ethics is a difficult fit to the analysis of algorithms because philosophically a virtue ethic represents a disposition rather than a rule. An "honest" algorithm in virtue ethics would be one that behaves honesty in a wide variety of situations, some of them difficult to foresee. A virtuous person is not a person who rigidly

[14] A strict consequentialist would be a monster, as consequentialism "seemingly demands...[that] innocents be killed, beaten, lied to, or deprived" for the greater good." (Alexander & Michael 2012).

applies a rule regardless of the context, indeed "rigid" is a vice. Even if we accept that an algorithm can have a disposition or a character, it has been argued that intentionality or "mindset" is the key component of virtue ethics, and while we may grant that a non-human algorithm has agency and power, granting it intentionality (or a mind) seems a bridge too far (Latour 2005 p. 71).

This means that when we describe an algorithm as "fair" we probably mean to reference either its consequences or (possibly) some absolute norm of fairness, and not that the algorithm itself has innate virtue. Virtue ethics as an avenue of reasoning about algorithms does not seem particularly plausible.

Consequentialism and Context: The Easiest Path

We are then left with consequentialist and deontological ethics. Consequentialist ethics asks us to turn to the results in a particular context to judge morality -- it asks us to consider consequences. Considering the ethics of an algorithm by considering its consequences demands that we trace its instantiation in actual situations, or at least imagine them. We must therefore understand the web of associations and interconnections that algorithms have with other technologies and people. As the same algorithm might have very different consequences in different situations, this might seem to present a problem for pursuing any ethical reasoning that focuses on the algorithm itself.

Luckily, this challenge is easily surmounted. While indeed the same algorithm might have different consequences in different situations, the same algorithm might also have the same consequences. That is, there may indeed be broad trends in the consequences of certain algorithms and it may be crucial for society that we detect them. In the scholarship that is normatively critical of particular algorithms (e.g., Introna & Nissenbaum 2000; Gillespie 2012) there is no sense that these authors expect the algorithm to "operate independently." A focus on the algorithm does not preclude attention to the context, nor does it substitute for nuance. In fact, this movement is arguing that within

the nuanced technical details of an algorithm we find important political consequences that have not received public attention commensurate with their societal importance (Introna & Nissenbaum 2000: p. 181). No doubt the recognition that every technical system operates within an interconnected and complex web of context is a penetrating diagnostic strategy, but--we emphasize--in addition it is also a useful strategy to learn to read and investigate algorithms as a particularly prevalent and potentially significant component of our evolving infrastructures.

A short example will help to cement the consequentialist case and also explain that focusing on the technical details of the algorithm is not inconsistent with considering systems and consequences broadly. Crain (2013) considers the late 1990s evolution of two broad classes of algorithms that are intended to determine which advertisements to display to particular users on the Web. He explains that *profile-based* algorithms employed by DoubleClick attempted to uniquely identify particular users, linking their online activity to their offline life. Once this link was established, DoubleClick would then mine a user's actual demographics and past purchases (from credit card statements) for intent to purchase certain products and display related ads. In comparison, *interest-based* algorithms employed by Google attempted to mine the keyword search history associated with a particular Web browser to infer purchase intentions.[15]

DoubleClick's *profile-based* algorithms relied fundamentally on amassing immense databases of uniquely-identifying information, giving DoubleClick's approach far more potential for some forms of negative consequences than Google's *interest-based* algorithms. These possible consequences include an increased risk of identity theft, the invasion of privacy, and civil rights violations. But understanding the algorithm in this case does not preclude a broader perspective--in fact it requires one. Crain points out that the decision to use one or the other class of algorithm was a business strategy that depended almost entirely on a company's access to inputs (large databases of personal information vs.

[15] "profile" vs. "interest" are industry terms in wide use for historical reasons. Of course the "profile" algorithms were intended to infer interests and the "interest" algorithms could be conceptualized as profiles.

large databases of search keyword history).

The distinction between the consequences of the two was significant for about five years, however, advertising companies then merged both approaches together, in part by using corporate acquisitions to secure the inputs for their competitor's algorithm. (Google acquired DoubleClick in 2007.) Although the "profile vs. interest" distinction is no longer common in the industry, it remains valuable to understand the technical process by which advertisements are selected, as the current hybrid approach retain the worst dangers of its predecessors. Understanding the business context is certainly important in Crain's analysis, but so is understanding the details of the algorithms--indeed the business strategy doesn't make sense without them. Crain does not need to resort to considering algorithms as immutable or isolated.

Of the three kinds of ethical reasoning considered here, consequentialist ethics seems the most prevalent way to think about algorithms, and it is an intuitive way to think about an ethical problem, akin to asking, "What will happen if we do it this way?"

Thinking Deontologically: Locating Algorithmic Norms

In contrast, the third form of ethical reasoning -- a deontological ethics of algorithms -- feels like a counter-intuitive proposition. Deontological honesty is neither the character trait "honest" nor a decision made after considering the consequences of lying. Instead, it requires the development of a series of moral and potentially even absolute rules ("do not lie" or "do not duplicate copyrighted software without permission") that can be followed. The word itself is derived from the Greek for "duty." Deontological ethics for algorithms feel unpromising largely because the deontological precepts that are easiest to grasp without resorting to consequentialism seem unlikely to appear in the realm of the algorithm as we define it in the next section. Some ideas that are likely justified deontologically (they don't necessarily depend on consequences) and are widely accepted as reasonable include

pacifism, "treat your family better than strangers," "*X* is forbidden by God," and the statement "torture for any reason is wrong." As our existing norms are geared toward human behavior it isn't immediately clear how this would be helpful to analyze algorithms. We could agree that torture is always wrong but algorithms seem unlikely to torture.[16] Therefore, the analyst commonly turns to consequentialism.

But pushing this argument further we contend that it is an interesting and open intellectual question as to whether a set of norms exists or could exist for algorithms. Applied ethics in real-world settings typically incorporates both rule-based and consequences-based reasoning. Indeed, as we explain below, it can be very useful to explicitly rule out some kinds of algorithmic process *a priori.* We contend that the reason we don't have such ethical rules now may be that the question has not been considered, rather than such rules are not possible. Additionally, just as it is often held that torture is wrong regardless of the situation, or certain human rights are absolute, having an absolute rule for algorithmic behavior does not foreclose our reasoning about consequences in addition to specifying rules in advance.

To consider algorithms deontologically -- that they must comply with a set of norms or moral duties -- is intellectually interesting in part because it has the potential to provide guidance for algorithm designers. In this framing, the command: "Comply with published protocols!" could be ethical guidance as well as practical engineering advice. Indeed, this may already be the case. An algorithm for sending SPAM could then be unethical regardless of whether or not it successfully sends SPAM, and irrespective of the consequences of SPAM. It could be judged unethical deontologically because it performs e-mail spoofing in violation of the SMTP protocol ("Comply with published protocols!") which governs e-mail. While most rules in the SMTP protocol appear there to make the overall system of e-mail function correctly, other rules and explicit reference in the protocol to

[16] To be clear, this isn't because machines can't torture, but because an algorithm as we define it is such a low-level abstraction in computing that it would be difficult to connect an abstraction "algorithm" (like "*a* + *b*") to the act of torture. This is discussed further in the previous section which defines algorithms.

"spammers" have a moral valence and seem justified by the fact that some messages are desirable and others are undesirable--a function of community norm rather than efficiency (Brunton 2013).

In general, although reasoning deontologically about algorithms seems intuitively difficult this is an area of interesting work where little analysis has been attempted so far. It is our conclusion, as we hope the examples in this paper have demonstrated, that deontological ethics of algorithms should not be discounted and that some *a priori* ethical algorithmic norms might be possible. In the case of a hypothetical surveillance system described above, the use of a judgement about race in a surveillance algorithm is already part of a broader public moral conversation about how we should think about race. To forbid race in the process of making college admissions decisions and in the process of surveillance are both deontological positions.

Conclusion: Focusing on the Algorithm, Narrowly Defined

We have argued above that is essential to be essentialist about algorithms. Indeed, to say that algorithms are a usefully defined class of things that constitute an important point of intervention can be likened to talking to a critic of ethics and technology writ large in 1945 and noting that nuclear weapons might bear looking into. To be clear, a *binary search* algorithm might not produce the next Hiroshima, yet it isn't impossible to imagine that "algorithm" could become one of the iconic technologies of the contemporary world—with consequences that are similar in scale to the automobile, the gene, or the atomic bomb. At least for now algorithms represent a fruitful path of investigation, with important dynamics and consequences that could deserve their own dedicated analytical vocabulary.

In the preceding text we have often proceeded by endorsing the idea, at least by implication, that algorithms have fixed characteristics. We believe this can be a powerful analytic strategy, and indeed it is a crucial one if we are to engage as normative analysts of contemporary technological

infrastructure so that we have some handhold on what it is we are investigating and arguing about. In this paper we have tried to construe the term "algorithm" narrowly and specifically, to relate it to ethical reasoning, and we have used some algorithm examples concerning race and surveillance in an attempt to demonstrate that it is possible and hopefully practical to ask ethical questions by focusing on algorithms as even though they are often technical, black-boxed, "low-level" processes in computing.[17]

To conclude we now return to a discussion of the role of algorithms generally as opposed to reasoning about particular instantiations or kinds. This constitutes our argument for why it is relevant to focus on algorithms rather than other components of a system, which as we have already admitted, are important as well. Our infrastructures, as others have pointed out, are increasingly networked and dominated by computation. The remarkable new prevalence of algorithms and the widespread claims for their significance are reason enough to investigate their consequences. However, in addition, there are several other reasons. We will discuss three.

The Increased Delegation of Authority

First, as they are implemented today, algorithms have a remarkable ability to *delegate authority*. Technology itself could be described as a way to delegate authority or control (e.g., Pinch & Bijker 1984), but even beyond this generality for all technology the algorithm has fostered this delegation to great effect. A remarkable amount of apparatus has emerged in the last 50 years to standardize and disseminate algorithms and programs (from computer science education to GitHub). It is this apparatus that makes the "black boxing" of algorithm implementations as interchangeable parts possible. This means that algorithmic design, intent, action and consequence can be distant in space and time. When authors describe "algorithmic infrastructure" they intend to highlight systems that operate with algorithmic logic, but "the infrastructure of algorithms" is just as important as it leads to

[17] Or, if considered in terms of abstraction rather than obscurity, an algorithm could be more "high-level" than what we normally consider.

important changes in how our sociotechnical systems function.

As we have argued that the details of algorithmic process are important, we have intended to highlight that choosing one algorithm over another in order to address the same problem may be a decision of significance. So might the manner in which an algorithm is implemented. Of course, circumstances also constrain the choice of algorithm, or even its design--as Google pushed for better algorithms to infer purchase intent from search keywords because as a search engine company it had ready access to a vast trove of search engine keyword data. Due to this delegation, an algorithm developed in a distant country may be the site of inquiry that a thorough ethical analysis of a technical system demands.

A Dangerous Dynamism

In our opening example about the history of photography, recall that we contrasted HP's "racist" face localization algorithm with the early history of photography and cinema. We pointed out that Dyer mounted an effective critique of photographic technology without the concept of algorithms. Although the term "algorithm" did not yet exist, Dyer's cameras presumably had algorithms in the sense that they had "processes," if not in a formal, narrow sense. These processes could be, for example, chemical. Yet it is today's implementation of algorithms in networked computers that asks us to reconsider our methods in the analysis of technology.

The distribution and reconfiguration of algorithmic processes implemented in computers can be much faster than most previous approaches to distributing control, logic or organizing process in technology design. This is one of the reasons computation is exciting and the fact that our newer technologies are networked and reconfigurable is widely seen as a benefit. When implemented in the computer, the algorithm can often be changed at remarkable speed. As Steiner (2012) points out when discussing catastrophic examples of automated trading algorithms used by Wall Street, an algorithm's

consequences can also be extremely rapid. The HP face localization problem was changed via a software update that modified the operation of thousands of cameras that had already been deployed, and this software update was deployed within a few weeks.

While an algorithm doesn't necessarily have to be instantiated in a way that is easy to update (as discussed above), the fact that this capability is possible changes the functioning of our technological apparatus profoundly. The metaphor of the technology's diffusion changes from that of an inventor hopefully sending products out into the world to that of a puppeteer whose movements remain always connected to his performance.

Stabilization and the Algorithm's Differential Obscurity

In the study of science and technology, *stabilization* (for a review, see Humphreys 2005) has been an important concept used to explain the evolution of technologies and ideas over time. It refers to the point at which a single technological artifact becomes the default representation of that kind of artifact in the minds of either technology developers or the public. The concept developed, in part, from scholars of science who tried to understand how scientists determined which areas of work contained important unsolved problems and which were closed. Although the term was meant to refer to physical artifacts in technology studies, we can see from the discussion above that it is an important analytical question to examine the degree to which an algorithm has been stabilized. The ease or difficulty of that stabilization is another relevant area of inquiry.

To explain, some technologies appear to employ the potential re-programmability of the algorithm (such as the Google search algorithm or the Twitter "trending topics" algorithm) to make a key feature of their system modular and subject to continual revision. While modularity and interchangeable parts has been a critical concept in the history of technology since the industrial revolution, if software were explained using the metaphor of a car, the people at Google and Twitter

don't just want to be able to use different tires in winter, or obtain transmissions from a different supplier, they want to change the engine every day. This has focused attention on the plasticity of algorithms, as mentioned in the previous point, and suggests a kind of *permanent destabilization* for some algorithms (or at least long-term destabilization).

Scholars of these algorithms have called them opaque, but this refers to corporate secrecy and the complexity and unfamiliarity of math and computer code. Yet this plasticity is only one possible use of algorithmic logic and this is only one kind of opacity.

Recalling our earlier point about the distribution of authority, imagine, for instance, that the Viola-Jones Recognizer becomes the accepted, normal algorithm for detecting the presence of faces. We know that algorithms often contain other algorithms and face detection is only one step among many for a surveillance application. This presents the possibility that an ethically problematic algorithm, once *stabilized* or widely accepted as the default solution for a particular problem becomes encapsulated in an encapsulation. It can be buried so deeply in the logic of a system that it might be very difficult to detect, even for engineers working on the system--who might not know that they use it.

While the role served by that particular localization algorithm might be made visible and it might be made modular, it can also be designed into expensive-to-change hardware, buried, and forgotten. This again argues for the relevance of algorithmic skills that allow a facility with the relevant ideas in math and computer science, as well as the education of computing professionals in social science and ethics. At base, this may not be a different tactic than the cinema historian (Dyer) learning about the chemistry of photographic processes. Yet saying that today's ethical critic needs a facility with these tools of computing imagines a different kind of scholar that is able to bridge the social and technological.

Sources Cited

Alexander, L. & Moore, M., (2012). "Deontological Ethics", In: E. N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy*. http://plato.stanford.edu/archives/win2012/entries/ethics-deontological/

Angwin, J. (2013). "The 'What They Know' Series." Available: http://juliaangwin.com/the-what-they-know-series/

Barocas, S. Hood, S., Ziewitz, M. (2013). Governing Algorithms: A Provocation Piece. Paper Presented to "*Governing Algorithms*." May 16-17, New York University.

Bermejo, F. (2007). *The Internet Audience: Constitution and Measurement*. New York: Peter Lang Publishing.

Bowker, G. C. & Star, S. L. (1999). *Sorting Things Out: Classification and Its Consequences*.

Bonilla-Silva. E. (2009). *Racism without Racists: Color-Blind Racism and the Persistence of Racial Inequality in America*. New York: Rowman-Littlefield.

Brunton, F. (2013). *SPAM: A Shadow History of the Internet*: Cambridge, MIT Press.

Callon, M., Law, J. & Rip, A. (eds.) (1986). *Mapping the Dynamics of Science and Technology*. London: Macmillan

Chetverikov, D. and Lerch, A. (1993). “Multiresolution Face Detection,” *Theoretical Foundations of Computer Vision*, vol. 69, pp. 131-140.

Crain, M. (2013). *The Revolution Will Be Commercialized: Finance, Public Policy, and the Construction of Internet Advertising*. (Unpublished doctoral dissertation.) University of Illinois at Urbana-Champaign, Urbana, IL.

Diakopoulos, N. (2014, February). “Algorithmic Accountability Reporting: On the Investigation of Black Boxes.” *Tow/Knight Briefs*. Tow Center for Digital Journalism, Columbia University. Available: http://www.nickdiakopoulos.com/wp-content/uploads/2011/07/Algorithmic-Accountability-Reporting_final.pdf

Dyer, R. (1997). *White*. Routledge: New York.

Fleck, M.,Forsyth, D., and Bregler, C. (1996) ``Finding Naked People,'' 1996 European Conference on Computer Vision , Volume II, pp. 592-602.

Floridi, L. (1999). Information ethics: On the philosophical foundation of computer ethics. *Ethics and Information Technology 1*: 37-56.

Fuller, Matthew (ed) (2008). Software Studies: a Lexicon. Cambridge: MIT Press.

Galloway, A. (2006). *Gaming: Essays on Algorithmic Culture*. Minneapolis: University of Minnesota Press.

Gangadharan, S. (ed.) (2014). “Data and Discrimination: Collected Essays.” Washington, DC: The New America Foundation Open Technology Institute. Available: http://www.newamerica.org/downloads/OTI-Data-an-Discrimination-FINAL-small.pdf

Gillespie, T. (2012). Can an Algorithm be Wrong? *Limn*. http://limn.it/can-an-algorithm-be-wrong/

Graf, H. P., Chen, Petajan, E. and Cosatto, E. (1995). “Locating Faces and Facial Parts,” *Proc. First Int’l Workshop Automatic Face and Gesture Recognition*, pp. 41-46, 1995.

Grimmelmann, J. (2009). The Google Dilemma. *New York Law School Law Review 53*: 939-950.

Hamilton, K., Karahalios, K., Sandvig, C. & Eslami, M. (2014). "A Path to Understanding the Effects of Algorithm Awareness." In *CHI Extended Abstracts on Human Factors in Computing Systems* (alt.CHI). ACM, New York, NY, USA, 631-642.

Hannak, A., Sapiezynski, P., Kakhki, A. M., Krishnamurthy, B., Lazer, D., Mislove, A., and Wilson, C. (2013). Measuring Personalization of Web Search. Proceedings of the 22st International World Wide Web Conference, Rio de Janeiro, Brazil, May 2013. Available: http://www.ccs.neu.edu/home/cbw/pdf/fp039-hannak.pdf

Humphreys, L. (2005). Reframing Social Groups, Closure, and Stabilization in the Social Construction of Technology. *Social Epistemology 19*(2-3): 231-253.

Latour, B. (2005). Reassembling the Social: An Introduction to Actor-Network Theory. New York: Oxford University Press.

Levin, G., Hayhurst, L., Benders, S., and White, F. (2008). Double-Taker (Snout). http://www.flong.com/projects/snout/

Introna & Nissenbaum – google algorithms paper 2001

Neyland, D. (2013). Bearing Account-able Witness to the Ethical Algorithm. Discussion Paper Presented to "*Governing Algorithms*." May 16-17, New York University.

Pasquale, F., (2014). *The Black Box Society: The Secret Algorithms That Control Money and Information*. Cambridge, MA: Harvard University Press.

Pinch, T. J. & Bijker, W. E. (1984). "The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other." *Social Studies of Science* 14: 399-441.

Sandvig, C. (2014). Seeing the Sort: The Aesthetic and Industrial Defense of "The Algorithm." *Media-N Journal 10*(3).

Sandvig, C., Hamilton, K., Karahalios, K., & Langbort, C. (2014). "Auditing Algorithms: Research Methods for Detecting Discrimination on Internet Platforms." Paper presented to "Data and Discrimination," a pre-conference of the 64th annual meeting of the International Communication

Association, Seattle, WA, USA. Available: http://www-personal.umich.edu/~csandvig/research/Auditing%20Algorithms%20--%20Sandvig%20--%20ICA%202014%20Data%20and%20Discrimination%20Preconference.pdf

Simon, M. (2009, December 23). HP Looking into claim webcams can't see black people. *CNN*. http://www.cnn.com/2009/TECH/12/22/hp.webcams/index.html

Slack, J. D. & Wise, J. M. (2005). *Culture and Technology: A Primer*. (3rd ed.) New York: Peter Lang.

Steiner, C. (2012). *Automate This: How Algorithms Came to Rule Our World*. New York: Portfolio.

Teodosio, L. and Bender, W. (2005). Salient Stills. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP). 1(1), pp. 16-36.

Wangsness, T., & Franklin, J. (1966), " "Algorithm" and "Formula" ", Communications of the ACM 9(4) (April): 243.

Yang, M-H., Kriegman, D. J., Ahuja, N. (2002). Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(1).

Acknowledgement

This work was funded in part by the Office for the Vice Chancellor of Research Interdisciplinary Innovation Initiative at the University of Illinois. The authors would like to thank Daniel Neyland and Mike Ananny for feedback on this work, as well as Darren Stevenson for editorial assistance.

Note

A previous version of this paper was presented to the "Governing Algorithms" symposium at New York University on May 17, 2013.