AutoCharge: Automatically Charge Smartphones Using a Light Beam

Yunxin Liu, Zhen Qin, Chunshui Zhao
Microsoft Research, Beijing 100080, China

Abstract

Smartphone charging imposes a big burden to users because they often have to recharge their smartphones every day or even multiple times per day. In this paper we try to answer the following question: can smartphones get automatically charged without requiring explicit effort from users? To this end, we propose a new approach, called AutoCharge, to explore the feasibility of automatic smartphone charging. The AutoCharge approach automatically locates a smartphone on a desk and charges it in a transparent matter from the user. This is achieved by two techniques. First, we leverage solar charging technique but use it in indoor spaces, to remotely charge a smartphone using a light beam without a wire. Second, we employ an image-processing-based technique to detect and track smartphones on a desk for automatic smartphone charging. As a result, AutoCharge is able to largely reduce users' efforts in smartphone charging and significantly improve the user experience. We have designed and implemented a prototype system of the AutoCharge approach. We report the design details of the light charger and the smartphone detection and tracking system. Experimental results show that our prototype is able to detect the presence of a smartphone within seconds and charge it as fast as existing wired chargers, demonstrating the feasibility of automatic smartphone charging.

1. Introduction

Today's smartphones are very power hungry. They use powerful hardware including multicore CPU, many GPU cores, large screen and high-speed wireless network interfaces, all with a high power consumption. They also run many energy-expensive applications such as high-end games, full HD video playback, and various continuous sensing tasks for context-awareness [2, 3 4]. As a result, many users suffer from a short battery lifetime on their smartphones and thus they often have to recharge their smartphones every day or even multiple times per day.

The frequent smartphone recharging imposes a big burden to users. As people increasingly depend on their smartphones for daily work and life, running out of battery becomes a very unacceptable situation for many users. To avoid such an unpleasant situation, users must keep a careful eye on the battery status of their smartphones and manually connect a charger to charge their smartphones when the battery is low. Doing so every day not only consumes a lot of user attentions but also imposes a mental burden to users.

Ideally, smartphones should automatically get recharged so that users do not need to worry about recharging their smartphones. However, existing solutions cannot achieve this desirable goal. With a wired charger, users must manually plug the charger into a smartphone, to explicitly express their intention of charging. Even with a wireless charging pad (e.g., the one used by Lumia 920 [15] or Nexus 4 [16]), users still need to explicitly put a smartphone onto a small charging pad to indicate the intention of charging. That is, users still have to manually connect a charger to their smartphones, resulting in the similar user burden as wired chargers. Therefore, wireless charging pads do not improve the user experience much.

In this paper, we propose a new approach, called AutoCharge, to improve the user experience in charging smartphones. Instead of forcing a user to explicitly indicate the intention of charging by plugging in a wired charger or contacting a small wireless charging pad, in our AutoCharge approach, the charger itself identifies the opportunities of charging a smartphone when a user puts the smartphone onto a desk. The charger automatically locates the smartphone and starts to charge it if the battery is low, without requiring any explicit effort from the user (see more details in Section 2). Consequently, the user is able to largely get rid of the burden of the manual smartphone charging and the user experience is significantly improved.

Our AutoCharge approach takes advantage of two kinds of key techniques to enable automatic smartphone charging. First, we leverage mature solar charging technique but use it in indoor spaces. Solar charging is able to remotely charge smartphones without a wire and thus is promising to improve the user experience of smartphone charging. However, solar charging is not widely used on smartphones because of several limitations. It works only in outdoor spaces but not indoor spaces. Due to severe scattering, the indoor surrounding light is usually much weaker (two orders of magnitude weaker or even worse) than the sunlight and thus cannot be used to charge a smartphone. Even when users are in outdoor spaces, they usually put they smartphones in a pocket or a bag so that solar charging cannot help. Furthermore, solar charging heavily depends on the weather condition and time. Solar charging does not work at night or when it is cloudy or rains. By applying solar charging to indoor spaces, we make it work for 24 hours per day, no matter what the weather condition is. To meet the requirements of charging a smartphone, we design the charger to generate a straight light beam with little scattering so that we can improve the charging efficiency and speed.

Second, we design a system of smartphone detection and tracking for automatic smartphone charging. The charger uses a camera to keep monitoring a surface such as a desk in office. When a smartphone is put onto the desk, the charger can quickly detect the presence of the smartphone. If the smartphone is able to be charged by a light beam and its battery is low, the charger starts to charge the smartphone. The charger uses a rotating motor to adjust the direction of its light beam so that it is able to accurately project the light beam on the smartphone. After the battery is full, the charging stops. The whole process is completely automatic and transparent from the user. In addition, the system is able to support multiple smartphones. If there are multiple smartphones on the same desk, the charger can charge them one by one.

We have developed a prototype implementation of AutoCharge. Experimental results show that our prototype system is able to automatically detect a smartphone from a picture in 0.3 seconds and projects its light beam onto the smartphone within one second to starts charging. The charging speed is comparable with wired smartphone chargers and may be further improved.

To the best of our knowledge, AutoCharge is the first work for automatic smartphone charging using a light beam. The main contributions of this paper are as follows.

- We propose a new smartphone charging approach called AutoCharge. By using a light beam, the AutoCharge approach is able to charge smartphones remotely without a wire or contact. We address various practical issues in using light charging in indoor spaces, paving the way for automatic smartphone charging.
- We design a camera-based system for automatic smartphone detection and tracking. We develop an algorithm for fast smartphone detection from pictures. Combining with light charging, we enable automatic smartphone charging which significantly reduces user effort in charging smartphones.
- We report the design, implementation and evaluation of the light charger and the smartphone detection and tracking system. Experimental results demonstrate that it is feasible to build an automatic charging system using a light beam.

The rest of the paper is organized as follows. In Section 2, we describe how the AutoCharge approach works and the targeted usage scenarios. We present the design of the light charger in Section 3 and the details on how to detect and track smartphones in Section 4. We report our prototype implementation in Section 5 and the evaluation results in Section 6. We discuss the limitations of our current implementation and future work in Section 7. We survey the related work in Section 8 and conclude in Section 9.

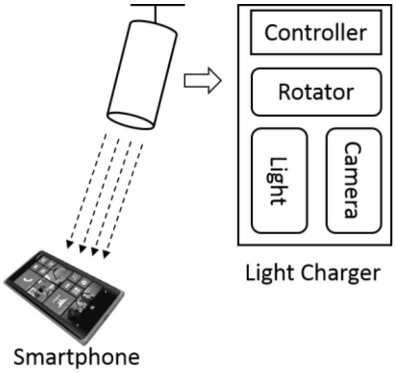


Figure 1: Illustration of the AutoCharge approach.

2. System Overview and Usage Scenarios

2.1 System Overview

Figure 1 illustrates how a system of AutoCharge works. The system consists of two parts: a light charger and a smartphone. The light charger has four components: a light which generates a straight light beam; a camera which monitors a surface like a table; a programmable rotator which moves the light and the camera to adjust their direction; and a controller which controls the movement of the rotator and turns on/off the light. The controller also runs necessary software to analysis the images captured from the camera to detect a smartphone. To be charged by the light charger, the smartphone must integrate a solar panel to harvest energy from the light beam of the light charger.

The light charger works in two modes: a *detection* mode and a *charging* mode. In the detection mode, the light is turned off. The camera is on and continuously takes pictures. The captured pictures are sent to the controller which analyzes the content of the pictures to decide whether there is a smartphone or not. The rotator periodically moves the direction of the camera to scan a large search area. If a smartphone is detected, the rotator stops and the light charge goes to the charging mode.

In the charging mode, the light is turned on. The camera continues takes pictures so that the controller is able to decide whether the light beam is correctly projected on the smartphone by analyzing the content of the pictures. If necessary, the controller controls the rotator to adjust the direction of the light beam to ensure that it is well received by the smartphone. If the smartphone is able to be charged by the light charger (i.e., it has the solar panel required by AutoCharge) and its battery is low, then the light charger starts to charge the smartphone. Otherwise, the light charger turns off the light and goes back to the detection mode. After the smartphone is fully charged, the light charger stops charging and switches to the detection mode. More technical details can be found in Section 4.

2.2 Usage Scenarios

The above system of AutoCharge may be used in various scenarios. Below we describe three of them.

Charging table. With AutoCharge, we can turn an existing table into a charging table. For example, we can install a light charger of AutoCharge on a desk in office (just like a desk lamp). Then, the desk immediately becomes a large charging surface. Whenever a smartphone is put on the desk, the light charger can quickly locate the smartphone and start charging it if needed. Note that the user does not need to put the smartphone on the desk explicitly for the purpose of charging. Putting a smartphone onto a desk is a natural action for many other purposes. For example, after checking an incoming short message or finishing a phone call, it is natural for users to put a smartphone onto a desk, particularly in office environment. Furthermore, even without using a smartphone, many users are used to put their smartphones on a desk after they come to office or go home. By automatically identify the opportunities of charging smartphones from existing user actions (i.e., putting a smartphone onto a desk), AutoCharge does not need any effort from users to express their intentions of charging a smartphone and to connect it to a charger. Consequently, AutoCharge is able to achieve effort-less smartphone charging and thus can significantly improve the user experience in charging smartphones.

Similarly, the charging table scenario also applies to home environments. However, two issues must be more carefully considered for AutoCharge to be used at home. The first issue is safety. Home environments are usually more complicated than offices. We must ensure that the light beam will not burn anything and more importantly, will not hurt anybody, especially kids. The second issue is that the light beam might be annoying at night, particularly if the light charger is used in a bedroom. We will show how to address the two issues in Section 3.

Charging room. By mounting a light charger of AutoCharge on the ceiling of a room, we can turn the room into a large charging space. The charger can locate and charge any smartphones on multiple tables or other surfaces in the room, increasing the chance to find and charge a smartphone, and making it more convenient to users. However, in this usage scenario, it may require a more careful design of the light charger. Due to the longer distance between the charger and a smartphone, the light beam needs to be even straighter and may use a higher power, and the rotator needs to be more accurate, compared to the charging table case.

Charging cube. As a mini version of charging room scenario, we can put a light charger of AutoCharge into a box (or build a dedicated one for AutoCharge) and turn it into a small charging cube. A user can easily throw a smartphone into the box to charge it. Due to the small space of the cube, it may be easier to design the charger and safety is a less critical issue because the light beam is contained within the box. However, this scenario requires users to explicitly put a smartphone into the box and thus is less user-friendly than the above two cases.

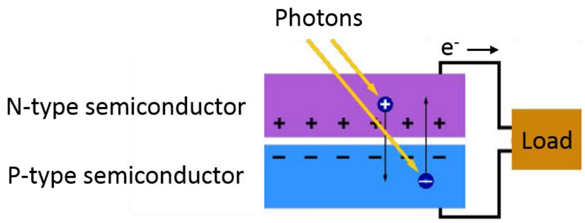


Figure 2: A PV cell generates electricity from light by the photovoltaic effect.

In this paper, we focus on developing a prototype system of AutoCharge for the *charging table* scenario but the most parts of the design also apply for the other two scenarios. Next in Section 3 and Section 4 we describe the design details of each component of AutoCharge and show how they work together to enable automatic smartphone charging.

3. The Light Charger Design

In this section we present how to design a light charger of AutoCharge. We first introduce how light charging works as the background. Then we explain why light charging is the right design choice compared to other alternative charging methods, followed by describing how to address various practical issues of light charging.

3.1 Background: Photovoltaic Charging

The device used to harvest energy from light is known as a solar cell, or more formally, a photovoltaic (PV) cell, because it converts the energy of light into electric energy by the *photovoltaic effect* [5, 6].

As illustrated in Figure 2, a PV cell consists of two types of semiconductor material: the positive type (p-type) semiconductor and the negative type (n-type) semiconductor. When the two types of material are put together, a *p-n junction* [6] is formed at the boundary of the two types of material and it creates an electric field. When the PV cell is exposed to light, the semiconductors absorb energy of the photons and release free electrons. The free electrons move according to the electric field and generate electricity. A PV battery or PV panel usually consists of an array of small PV cells.

Various types of semiconductor material can be used to build PV cells. The most prevalent bulk material for PV cells is crystalline silicon, including mono-crystalline silicon and polycrystalline silicon. There are also non-silicon PV materials, such as cadmium telluride and gallium arsenide. Different materials are of different cost and have different efficiency in converting light into electricity.

3.2 Why Use a Light Beam

Charging smartphones is inconvenient because that a user has to frequently connect a smartphone to a charger. A natural design choice to improve this is using contact-less charging, i.e., using a charger that is able to charge a smartphone *remotely*. We have explored two remote charging methods – using a light beam and using wireless power – and conclude that using a light beam is a better design choice than wireless power.

Wireless power is the transmission of electrical energy from a power source to a receiver without a wire conductor [7]. It is usually achieved through magnetic resonant coupling [8] using two coils or through electromagnetic radiation [9] which transmits energy by radio waves. However, wireless power methods have several disadvantages, preventing them from being used in our targeted usage scenarios. First, the electromagnetic radiation of wireless power is much higher than wireless communications (e.g., Wi-Fi or 3G). Thus, safety to human bodies is a big issue in wireless power. As a result, wireless power is usually used only in extreme scenarios such as in outer space, for military purposes, or in very short ranges. Existing wireless charging pads for smartphones are actually based on electromagnetic induction which is one of wireless power methods and only works within several centimeters. Second, as the radio frequencies used in wireless power are much lower than the frequencies of lights, it is hard to emit the radio waves within a straight beam. This causes energy waste if the receiver is not large enough and makes it hard to ensure safety. Using high frequency radio waves like x-rays can solve this problem but x-rays are harmful, leading to more severe safety issues. In addition, when used in a long distance (e.g., meters or longer), the power transmission efficiency of wireless power is low. Wireless power is still under active on-going research in terms of transmission efficiency, safety and transmission distance [10, 11, 12]. We believe that it takes more time for wireless power to become mature enough to be used in users' living spaces for long-distance power transmission.

Compared to wireless power transmission, PV cell techniques are much more mature and have been well proven in real deployments. Lights are easy to be generated, can be transferred in a straight beam through specular reflection or optical lens, and are less concerned in safety to human bodies. Note that essentially light is also a kind of electromagnetic wave. However, the underlying principle (i.e., the photovoltaic effect) of light charging is so different from electromagnetic radiation that light charging is treated as a different method.

3.3 Addressing Practical Issues

While light charging is promising for remote smartphone charging, some practical issues must be considered to use it in indoor spaces. Below we describe those practical issues and how to address them.

Efficiency. The first issue is whether the PV panel attached to a smartphone is able to generate enough current and voltage to charge the battery of the smartphone quickly. Due to the size limit, one cannot integrate a PV panel larger than a smartphone into the smartphone. This issue can be addressed from three aspects: 1) using a PV panel with a high efficiency of converting light into electricity; 2) using a high power light to generate a strong light beam; 3) using a proper optical lens to shape the light beam straightly to avoid energy waste caused by scattering. With the above methods, we are able to build a prototype system of AutoCharge which can charge smartphones as fast as existing wireless chargers, as we will show in Section 6.

Safety. As we use a light beam stronger than normal lights, safety issues must be carefully considered. First, we must prevent the light beam from hurting people, particularly eyes as a kid may be curious and try to watch the light beam directly. To achieve it, in the charging mode, the camera of a light charger of AutoCharge keeps monitoring during charging. If the camera detects there is any object blocking the light beam, the controller of the light charger will immediately turn off the light to ensure safety. Second, we must prevent the light beam from burns any things. As we will show in Section 6, the required energy level of the light beam in AutoCharge is similar to normal sunlight. It cannot burn a piece of paper even if the paper is continuously exposed under the light beam for hours. Furthermore, the majority of the light energy is absorbed by a PV panel and thus only a small part of the light energy is transformed into heat. In addition, as we have described in Section 2.1, AutoCharge keeps the light continuously on only if there is an AutoCharge-compatible smartphone which is in low battery. Thus, the light beam will not keep baking anything else.

PV panel integration. When a user puts a smartphone on a desk, the screen of the smartphone usually faces up. One may wonder how to integrate a PV panel into a smartphone so that the PV panel can receive lights even when the screen of the smartphone faces up. This issue can be solved by using a transparent PV panel. For example, Wysips [1] provides transparent PV films with pretty good PV efficiency which can fully charge a smartphone battery within several hours. Such a transparent PV film can be attached to the screen of a smartphone without affecting users in watching or interacting (e.g., through touch) with the screen, just like a screen-protection film. To take advantage of the AutoCharge approach, we imagine that future smartphones may integrate a PV film/panel on both sides. Therefore, no matter whether the screen of a smartphone faces up or down, the smartphone can always be charged by a light beam.

Light beam visibility. A light beam is usually visible. It can be annoying if a visible light beam is used in a bedroom at night. To solve this issue, we may use invisible lights. For example, we may use an infrared light beam which is invisible and as safe as normal lights (other kinds of invisible lights such as ultraviolet lights are not safe). Many PV materials including crystalline silicon and gallium arsenide are actually able to work with infrared lights. As a result, the AutoCharge approach can be even more user-transparent and just works silently without annoying users even at night when users are in sleep.

4. Smartphone Detection and Tracking

This section describes how we can automatically charge a smartphone without any explicit effort from a user, including how to decide when a smartphone should be charged, how to detect the presence of a smartphone, and how to project a light beam onto the smartphone to start charge it.

4.1 Deciding When to Charge

A light charger of AutoCharge should charge a smartphone only when the smartphone needs to be charged (i.e., the battery of the smartphone is low) and stop charging once the battery is fully charged. That is, the light charger must be able to know whether the battery of a smartphone is low or full. Furthermore, the light charger should not wrongly charge any objects which look like a smartphone or any other smartphones which cannot be charged by a light beam.

Challenge. Before a light charger is able to charge a smartphone, they must know each other. This is not straightforward because there is no longer a user who can help connect them together. A simple approach is to let a smartphone to send out beacons when its battery is low. For example, a smartphone may periodically broadcast a message using Bluetooth or blink a LED (light-emitting diode) light with a pre-set pattern to tell that its battery is low and it can be charged by a light beam. A light charger may recognize such a beacon through Bluetooth or camera. However, this approach does not work in practice due to two reasons. First, sending beacons causes extra energy consumption of a smartphone. It may take a long time (e.g., several hours) before a smartphone is put on a desk that has a light charger. If the smartphone keeps sending beacons, it may use up all its battery quickly. Second, this approach does not work when a smartphone is totally out of battery because it cannot send out any beacons. In fact, any approaches requiring a smartphone to send active information will suffer from the above two limitations, making it a challenging task for a light charger to find a smartphone and decide the status of the battery of the smartphone.

Solution. To address the challenge, we employ the following approach. When the battery of a smartphone is low, we do not require the smartphone to do any extra work. It just works as normal. When a light charger detects the smartphone (see Section 4.2 on how to do it), the charger first tries to charge the smartphone using a light beam. On the smartphone-side, together with a PV panel, we also integrate a microcontroller (e.g., MSP430 [27]) and a LED light. The microcontroller and LED light are powered by the PV panel rather than by the battery of the smartphone. When there is not a light beam, the PV panel does not generate any electricity, thus the micro controller and the LED light are powered off and do not consume any power. When the light charger turns on the light beam and projects it on the smartphone, the PV panel starts to generate electricity and the microcontroller and the LED light are powered on. The microcontroller then checks the battery status of the smartphone. If the battery is low, the microcontroller controls the LED light to blink in a pre-set pattern to indicate a *ready-to-charge* message. The light charger recognizes the LED blinking pattern (using its camera), knows that the smartphone's battery is low, and thus continues to charge the smartphone. If the light charger cannot detect the expected ready-to-charge message within a given time period (e.g., 10 seconds), it stops charging and moves back to the detection mode. Once the smartphone's battery is fully charged, the microcontroller controls the LED light to blink in another pre-set pattern to indicate a *charge-completed* message. The light charger recognizes the charge-completed message, stops charging, and enters the detection mode again. Figure 3 illustrates how this approach works, including the transitions between the detection mode and the charging mode.

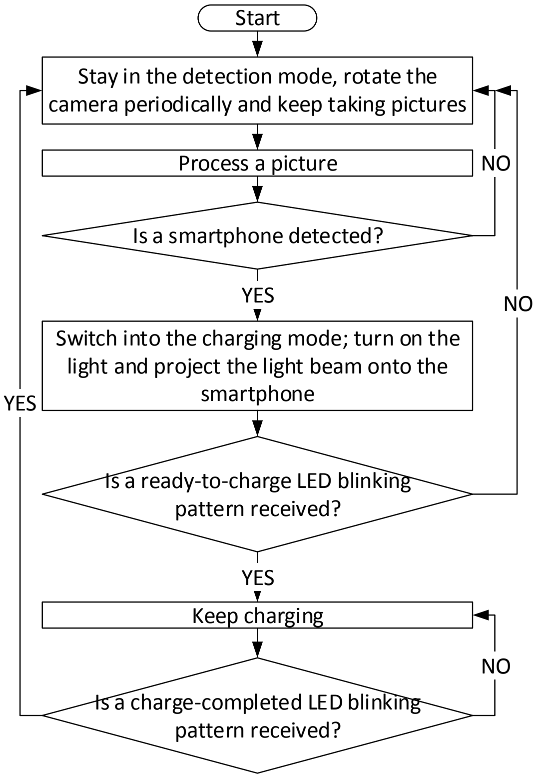


Figure 3: Work flow of a light charger of AutoCharge.

With the above approach, we ensure that the light charger will charge a smartphone only if it is designed to be charged by AutoCharge and it is in low battery. Even if the light charger wrongly detects a smartphone which is not compatible with AutoCharge or an object which looks like a smartphone but is actually not, the light charger will not charge them because they cannot generate the ready-to-charge LED blinking pattern. This approach also works for smartphones running out of battery. Once the PV panel of a smartphone starts to harvest energy from a light beam, the microcontroller and the LED light are powered up and thus can send out the ready-to-charge message. Consequently, AutoCharge is able to enable automatic and safe smartphone charging, without consuming any extra energy of a smartphone.

4.2 Detecting the Presence of a Smartphone

One key design principle of AutoCharge is keeping the changes of smartphone-side as minimal as possible. To this end, we design the charger to implement the complex and intelligent parts of smartphone detection. Specifically, we design the light charger to use a camera to detect the presence of a smartphone. By analyzing the content of pictures took by the camera, the charger may decide whether a smartphone is put on a desk.

We solve the smartphone detection problem as a problem of detecting a rectangle object with a proper range of size (i.e., height and width) from a picture. This is because most smartphones are roughly in a rectangle shape and detecting a rectangle object from a picture does not require any collaboration from the object. Therefore, this approach does not impose any burden on the smartphone-side. However, this approach may result in false-positive because not all rectangle objects are a smartphone. We handle this issue by using the ready-to-charge message as described in Section 4.1. Without receiving a ready-to-charge message, the charger turns off the light to avoid baking an object which is wrongly detected as a smartphone.

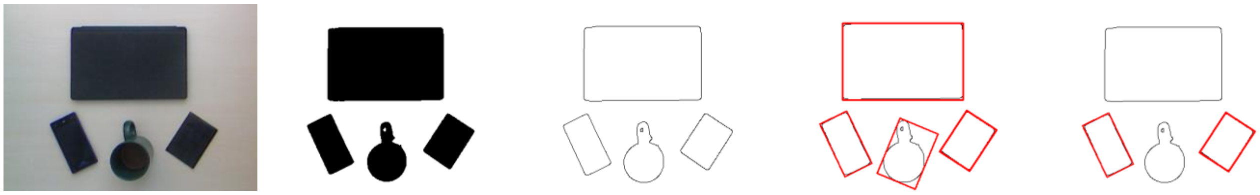
Challenge. As we cannot assume that a light charger has very powerful computation resources, we need to find a lightweight method for rectangle detection. This is not trivial. As an example, one method that we have tried is first extracting the straight lines in a picture and then constructing rectangles from the lines. Straight line extraction can be done using Hugh Transformation [13], a classic technique of line detection that has been widely used in digital image processing. However, we found that this approach does not work well. Because the edges of most real objects are usually not in ideally straight lines, Hugh Transformation may return hundreds lines from a simple picture (e.g., one with only several objects on a clean table) and even cannot find out the right rectangle objects. This makes the task of rectangle detection slow and challenging.

Solution. We use the algorithm shown in Figure 4 for fast rectangle detection. The algorithm takes a RGB (standing for *red*, *green*, and *blue*) picture as the input and works as follows. We first convert the picture into a grayscale image. Using the grayscale information, we further convert the image into a binary image. Then, we use the Border-Following approach [28] to extract the edges of the objects in the image. Each object's edges are represented in a set of points. After that, we compute the *convex hull* of each object's edges using the Sklansky algorithm [29]. The convex hull of an object's edges is the smallest convex polygon which encloses all the points of the object's edges. Sklansky's algorithm can be done in linear time and is much faster than Hugh Transformation. Note that even if we use Hugh Transformation, we also need to do the tasks of lines 4-6 in Figure 4. Once we get a convex polygon, we further compute the rectangle of the minimum area enclosing the convex polygon. We call such a rectangle as a min-rectangle of the corresponding convex polygon. If the area difference between the convex polygon and its min-rectangle is small (e.g., less than 20%), we conclude that the corresponding object of the convex polygon is a rectangle object and thus is probably a smartphone. By tolerating certain area difference, our algorithm is able to detect smartphones what are not in a strict rectangle. For example, many smartphones have rounded corners rather than straight ones. Our algorithm is able to easily detect those smartphones but the method based on Hough Transformation cannot.

```
 1: // Input: a RGB picture and its depth map;
 2: // Output: a set of rectangles;
 3: RectangleDetection()
 4: { Convert the RGB picture into a grayscale image;
 5:     Binarize the grayscale image into a binary image;
 6:     Extract object edges in the binary image;
 7:     Compute the convex hull of each object's edges;
 8:     Find out the min-rectangle of each convex hull;
 9:     Look up the depth values of each rectangle;
10:     Calculate the size of each rectangle;
11:     Filter out the rectangles that are too small or too large;
12:     Return the remaining rectangles; }
```

Figure 4: The algorithm to detect rectangles in a picture.

The next step is to decide the real size (i.e., the size in physical world) of the rectangles. To do it, we need to know the depth of the rectangles. The depth information can be obtained by using dual cameras with two different viewing angles or by using a depth camera such as the one used in Kinect sensor for Xbox 360 [14]. In the algorithm of Figure 4, we assume that we have the depth map of the input picture. With the depth map, we look up the depth values of each rectangle. Then we can calculate the physical height and width of the rectangle. Based on the sizes of typical smartphones, we can decide a range of size which covers most existing smartphones. Using the range of size, we can filter out the rectangles that are too small or too large to be a smartphone. Finally, we return the remaining rectangles as the candidates which look like a smartphone. Note that we



(a) original picture (b) binary image (c) edge extraction (d) min-rectangles (e) final outputs

Figure 5: Example of rectangle detection from a picture. From left to right: (a) the original picture took by the camera; (b) the binary image after binarization using the grayscale information; (c) the edges of the objects; (d) the min-rectangles of the objects; (e) the final outputs of two rectangle objects.

do not need a very accurate range of smartphone size, because we can tolerate false-positive in smartphone detection by using the ready-to-charge message.

Figure 5 shows an example to further illustrate how the rectangle detection algorithm works. There are four objects in the original RGB picture (Figure 5a): a smartphone (bottom-left), a tea cup (bottom-middle), a PV panel (bottom-right), and a tablet device (top). Except the tea cup, all the other objects are in a rectangle shape. After converting the picture into a grayscale image and doing binarization, we can clearly see the shapes of the objects in the binary image (Figure 5b). Using the Border-Following algorithm [28], we can extract the edges of the objects (Figure 5c). Then, using Sklansky's algorithm we can get the convex hulls of the objects and further compute the min-rectangles of the objects, as marked in red lines in Figure 5d. Next, by comparing the size difference between an object and its min-rectangle, we can decide that the tea cup is not a rectangle and filter it out. By comparing the size of an object to the pre-defined size range of smartphones, we can further exclude the tablet device because it is too large to be a smartphone. Finally, the algorithm returns two objects which are possibly a smartphone (Figure 5e): one is the real smartphone and the other is the PV panel. Clearly the algorithm wrongly treats the PV panel as a smartphone because that the size of the PV panel is similar to a smartphone. However, as we have mentioned, our AutoCharge approach is able to handle this by using the ready-to-charge message. For the purpose of illustration, here we exclude the tablet device. It is possible to extend the AutoCharge approach to charge other mobile devices such as iPad and Surface devices. Also for the purpose of illustration, the picture we used in Figure 5 is very simple: the table is clean the objects do not overlap. The real cases may be much more complicated. In Section 6, we will show the algorithm works well in various settings.

4.3 Charging a Smartphone

Once we get a rectangle which is supposed to be a smartphone, we turn on the light and try to charge the smartphone. This includes the following steps: 1) project the light beam onto the smartphone; 2) detect the ready-to-charge message; and 3) detect the charge-completed message. Furthermore, for safety consideration, the charger also detects whether there is an object blocking the light beam. In addition, we support charging multiple smartphones on the same desk.

Projecting the light beam. For efficient charging, we need to correctly project the light beam on the smartphone. This is achieved by using the rotator of the charger to adjust the direction of the light beam with the help of the camera. Similar to the smartphone detection described in Section 4.2, we use the camera of charger to locate the light spot of the light beam on the table. We first convert the picture captured by the camera into a grayscale image. Based on the grayscale difference in the image, we can determine the boundary of the light spot of the light beam in the image. Then, we calculate the central point of the area of the light spot and central point of the rectangle of the smartphone. After that, we control the rotator to adjust the direction of the light beam until the two central points are overlapped. Consequently, the light beam is correctly projected on the smartphone.

Detecting the ready-to-charge message. After the light beam is projected on the smartphone, the charger waits for a ready-to-charge message from the smartphone. We design the ready-to-charge message using the following LED blinking pattern: first keep the LED light ON for one second and then turn it OFF for 500 milliseconds. On the smartphone-side, after its PV panel starts to generate electricity and the LED light is powered on, the LED starts to repeat the above ON/OFF pattern for 10 times. On the charger-side, the camera of the light charger is used to detect the LED blinking pattern. If the ON/OFF pattern is detected for three times continuously, it is treated as that a ready-to-charge message is received and the charger decides that it is safe to continue to charge the smartphone. With a camera of 30 frames per second (fps), it can take a picture every 33 milliseconds. Thus, the ON/OFF pattern lasting for 1.5 seconds can be easily and reliably detected. And the detection can be done quickly within several seconds (three continuous patterns take 4.5 seconds).

Detecting the charge-completed message. Similarly, we design the charge-completed message as the following LED blinking pattern: first keep the LED light ON for two seconds and then turn it OFF for one second. When the smartphone's battery is fully charged, the smartphone repeats the ON/OFF pattern for up to 10 times. If the charger detected the pattern for three times continuously, it stops charging. Once the charging is stopped, the smartphone stops blinking its LED light as well. By using different ON/OFF durations, the charge-completed message and the ready-to-charge message can be easily distinguished from each other. The duration values can be further fine-tuned to reduce the message detection time.

Detecting a blocking object. If an object blocks the light beam, we should stop charging. In particular, there may be a curious kid who tries to look at the light beam directly and thus we must turn off the light as quick as possible. Detecting an object blocking the light beam can be easily done by observing the changes of the pictures from the camera. With a camera of 30 fps, we do the detection and turn off the light within 50 milliseconds, as we will show in Section 6. This is faster than human reaction for self-protection (e.g., in looking at the sun by incaution) and thus can ensure safety.

Supporting multiple smartphones. The AutoCharge approach naturally supports multiple smartphones. Once a light charger finishes charging one smartphone, it enters the detection mode again to search for another smartphone. If multiple smartphones are detected simultaneously (i.e., the algorithm in Figure 4 returns multiple rectangles), the charger charges them one by one without further searching.

5. Prototype Implementation

We have implemented a prototype system of AutoCharge. Figure 6 shows two pictures of our prototype implementation. For PV panels, we investigated three types of materials: mono-crystalline silicon (mono-Si), polycrystalline silicon (poly-Si), and gallium arsenide (GaAs). We found that the efficiency of the mono-Si and poly-Si panels is much lower than the GaAs panels and cannot be used to charge a smartphone. Thus, we chose to use GaAs. We designed a small circuit board to connect the PV panel to the battery of a smartphone. The small board consists of the necessary charging circuits, a LED light, and a MSP430 microcontroller [27] which controls the LED light to emit the ready-to-charge and charge-completed messages.

On the light charger side, to generate a light beam to charge a smartphone, we use an UltraFire CREE XM-L T6 Focusing LED Flashlight torch [30]. As shown in Figure 6, we took apart the torch and only used its head (i.e. the T6 LED Flashlight). We use a DFRobot FIT0046 DF15MG Tilt/Pan Kit [31] to control the direction of the light beam. The kit is able to rotate for 120 degrees in both horizontal and vertical angles in a speed of more than 350 degrees per second, with a maximum load of 15kg. It is programmable. We connect it to a PC and can control its movements in real time. For the camera system, we used a Kinect sensor [14] which has a RGB camera and a depth camera. The cameras take pictures in 640x480 pixels. We also connect it to a PC to program it.

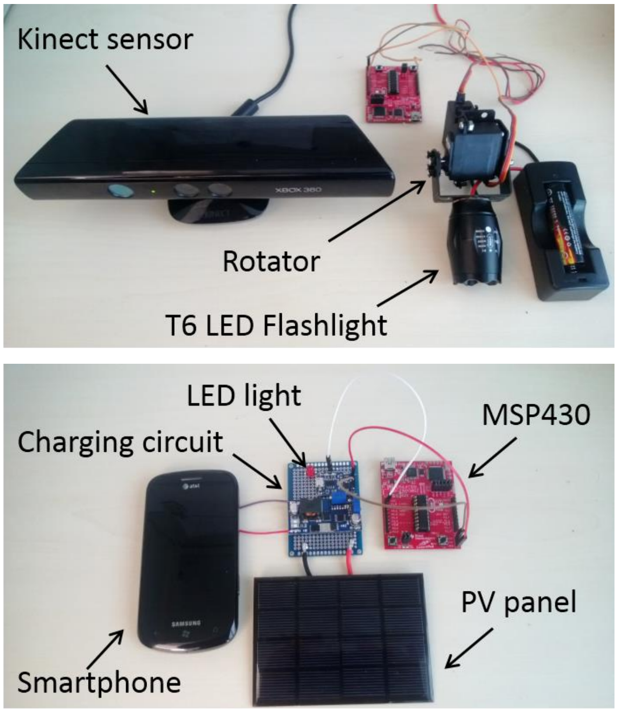


Figure 6: A prototype implementation of AuthoCharge. Top: the charger side components. Bottom: the smartphone side components.

We run all the software of the light charger on a HP6000 PC, including 1) detecting the presence of a smartphone, the ready-to-charge message, and the charge-completed message; 2) controlling the rotator to project the light beam on a smartphone; and 3) detecting a blocking object. The PC runs Windows 8 operating system with 8GB RAM and an Intel Core2 Quad CPU of 2.66GHz. We used the OpenNI [32], an open source SDK for 3D sensing to retrieve RGB images and their depth maps from the Kinect sensor. We used the OpenCV [33] libraries for the image processing tasks. In total our implementation has 1,520 lines of C++ code. In addition, implementing the Hough Transformation algorithm takes extra 670 lines of C++ code.

6. Evaluation

We evaluate AutoCharge using our prototype implementation by answering the following questions: 1) Can a PV panel of a size of a smartphone harvest enough energy from a light charger to charge a smartphone? 2) How accurately and quickly can a light charger detect a smartphone? 3) Once a smartphone is detected, how quickly can the light charger control the rotator to project the light beam on the smartphone? 4) Can the ready-to-charge message and the charge-completed message be reliably detected? 5) Is the system safe enough to be used in people's living environments like offices and homes?

Charging efficiency. We measured the efficiency of the GaAs PV panel in converting light into electricity. Under

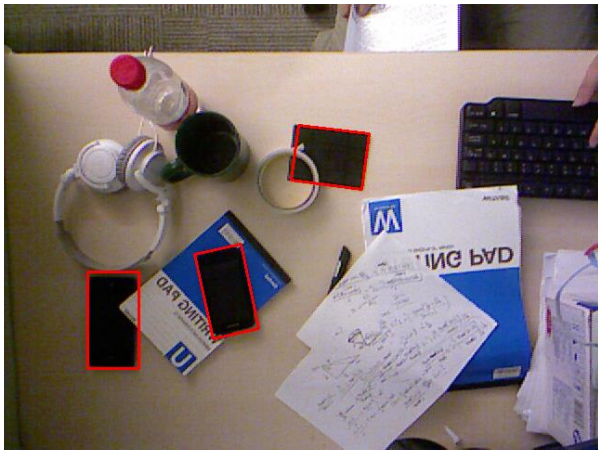


Figure 7: An example to illustrate that AutoCharge is able to correctly identify rectangle objects in a picture together with many other objects.

the light energy level of the T6 LED Flashlight, the output power of the GaAs PV panel is 37.1mW/cm^2. Consider a middle size smartphone such as a Lumia 920 which has a size of 13cm x 7cm, one may attach a PV panel of 91cm^2 to the smartphone. By using such a GaAs PV panel, we can roughly generate a power output of 3.38W. It is able to fully charge a battery of 2000mAH in 2.5 hours (assuming a charging voltage of 4.2V). Given that most existing wired smartphone chargers typically have a power output ranging from 2.5W (0.5A, 5V) to 5W (1A, 5V), these results show that it is able to charge a smartphone using a light beam as quickly as many wired chargers. If we consider the larger smartphones such as Galaxy NoteII which has a size of 15cm x 8cm, we can generate even more power.

Detection accuracy. We found that our implementation is able to correctly identify rectangle objects with various settings. For example, as shown in Figure 7, we can successfully detect the two smartphones (the two rectangles bottom-left, marked in red lines) and one other rectangle object (the rectangle up-right, marked in red lines) among many other objects, 1) no matter their locations and positions; 2) from different backgrounds (e.g., one smartphone is on top of a notebook; 3) even if they are partially overlapped with other objects (e.g., the left smartphone is covered by a corner of the notebook). This show that AutoCharge is able to work in typical office and home environments where people often put their smartphones on a table with other things.

However, we did find that our implementation could not work in several cases. First, if the background color is very similar to a smartphone's color (e.g., a black smartphone on top of a black laptop), we cannot detect the smartphone. Using more advanced image processing techniques may help solve this problem. Second, if the surrounding light is very dark (e.g. at night with all lights turned off in a room), our current implementation does not work. This issue may be solved by using the light charger to light up (using another weak light beam different from the one used for charging) the table during the detection mode. In addition, if a smartphone is largely covered by another object (e.g., under a notebook), our algorithm cannot find out the smartphone. In fact, this is a limitation of the AutoCharge approach because the light charger cannot charge the smartphone. However, this case rarely happens in practice.

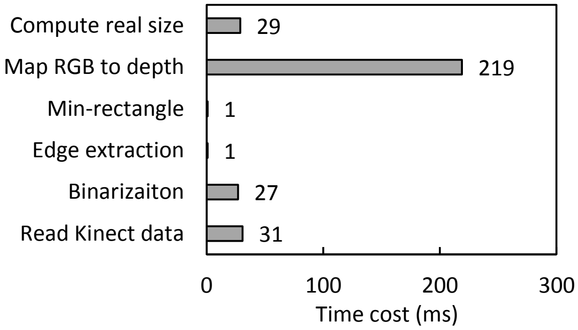


Figure 8: Time cost of each step in rectangle detection.

Detection time. Our implementation is able to quickly detect rectangle objects from a picture. We have measured the finish time of the function in Figure 4 using 10 pictures with various objects in different locations and positions and under different brightness of surrounding lights. The average result is only 0.31 seconds, with a range of from 0.27 seconds to 0.58 seconds. To cover a large table, we can rotate the camera of the light charger. For a large table of 4m^2 (2m x 2m), assume that the light charger is mounted 1m higher than the table, we need to take pictures from 4 different angles (i.e., rotate the camera for 4 times periodically) to cover the whole table. Each time of rotation takes about 0.5 seconds. Therefore, our implementation is able to detect a smartphone at most in 4.32 seconds, no matter where the smartphone is on the large table. For smaller tables, the number of rotation can be reduced or the camera even does not need to rotate at all and thus we can detect smartphones more quickly.

Figure 8 shows the time cost of each step of the rectangle detection algorithm. We can see that the most expensive step is "Map RGB to depth" which looks up the depth information for a given point of the RGB image. Due to the limitation of the API provided by OpenNI SDK, for a given point in the depth image, we can look up its position in the RGB image but we cannot do it oppositely. That is, for a given point in the RGB image, OpenNi does not provide an API for us to directly get its depth value in the depth image. Therefore, to decide the depth information of a rectangle in the RGB image, we first build a full RGB-depth map for all the 640x480 points of the RGB image, which is time-consuming. However, this can be optimized. If we implement our own function on depth-lookup rather than using the OpenNI API, we only need to compute the depth of a rectangle, which may significantly reduce the time cost, e.g., by an order of magnitude.

From Figure 8, we can see that other steps have a small time cost. In particular, the edge extraction and min-rectangle computation are very fast, taking only 2ms in total. It is much faster than the Hough Transformation which took 173 seconds on average (562 times higher) to processing the same 10 pictures (but with a larger resolution), ranging from 3 seconds to as long as 653 seconds. The time cost of the Hough Transformation heavily depends on the complexity (i.e., the number of lines) of the pictures. Furthermore, the Hough Transformation does not work for small pictures of 640x480. We had to use a large resolution of 1280x960 to make it work. Doing so reduces the frame rate of the Kinect sensor to only 12 fps. In addition, our implementation uses only 20MB memory.

Projection/rotation time. After a smartphone is detected, we need to quickly to project the light beam on it. This involves detecting the position of the light spot of the light beam and moving it onto the smartphone. We measured that our implementation is able to detect the light spot within 0.3 seconds (it does very similar work as rectangle detection). We are able to move the light spot onto the smartphone with just one movement, which takes about 0.5 seconds. As a result, projecting the light beam on a smartphone can be quickly finished within one second. Note that in our current implementation, the camera and the light are separated. In productization of the light charger, the camera and the light must be tightly integrated together and their relative positions can be pre-calibrated. The light charger may calculate the distance between the light spot and a smartphone and thus does not need to detect the position of the light spot, further reducing the projection/rotation time. The error of the rotator is less than 0.05 degrees. Assume that the light charger is 1m higher than a table of 2m^2, this leads to less than 2mm error in absolute distance in the worst case, which is acceptable in projecting a light beam.

We also experimented how large a table our current light charger can cover. Assuming the light charger is 1m higher than a table, with the rotation range of 120 degrees of our rotator, the cover area size is more than 6m^2. It is much larger than the normal size of typical tables. Thus, our implementation is able to enable the charging table scenario that we described in Section 2.2.

Message detection. The ready-to-charge message and the charge-completed message can be detected by our light charger very reliably. We experimented for 100 times to detect each of the messages. We got a success rate of 100% for the both messages. In fact, we were always able to detect the first three blinking patterns even message repeated its pattern for 10 times. The reason is very simple: our camera is able to take pictures in 30 fps but the blinking patterns take 1.5 seconds (for the read-to-charge message) or 3 seconds (for the charge-completed message), which are much longer than the time to take one picture (0.033 seconds). Note that the message detection times are not critical, particularly for detecting the charge-completed message. Compared to the total charging time of hours, several seconds latency in message detection are negligible. Thus, our design takes more considerations on detection reliability rather than short detection time.

Safety. We evaluate the safety of our implementation from two aspects. First, we measured the energy level of the light beam in our implementation. It is only 110 mW/cm^2 which is similar to the sunlight of AM1.5 spectral irradiance (about 100 mW/cm^2). It is actually much safer than sunlight because 1) the light beam is cool light and thus causes less heat than sunlight; 2) a large part of the energy of the light beam is converted by the PV panel into electricity and thus generates even less heat. Even if we keep expose the PV panel under the light beam for hours, the temperature does not increase much. We even cannot feel that the PV panel becomes warm using our hands. To compare, we can quickly feel that a smartphone becomes warm after using it for a while, e.g., playing a game. Therefore, the light beam is pretty safe and will not cause damages like burning a piece of paper or breaking a smartphone.

Second, we measured how quickly our light charger can detect an object that blocks the light beam and turn off the light. Using a camera of 30 fps, we can detect a blocking object within 50ms. To compare, when one looks into the sun by incaution, it takes more than 100ms for he/she to react by closing his/her eyes or turning his/her head. Therefore, our light charger will not hurt anyone even if a curious kid tries to directly look at the light beam.

7. Discussion and Future Work

While our evaluation results are encouraging, the AutoCharge approach also has some limits. For example, if a smartphone is under another object like a book, we cannot charge it. Furthermore, if one does not take out a smartphone from his/her pocket or bag, AutoCharge cannot help. In addition, when people travel to different places, it is unlikely they can find a light charger of AutoCharge in every room. However, as people spend most time in office and at home, AutoCharge may be used to largely reduce users' effort in charging their smartphones every day. In this paper, we focus on exploring new ways for smartphone charging and study the feasibility of automatic smartphone charging. We do not intend to completely replace the existing smartphone chargers using AutoCharge.

Another issue of the AutoCharge approach is that it does not work for existing smartphones because it requires that smartphone must integrate a PV panel to work with a light charger. However, smartphones are consumer electronics and we see that smartphone makers like Apple and Samsung typically release a new generation of smartphones every year or even less than one year. Many users also follow the short product circles of smartphones and upgrade their smartphones quickly, e.g., one or two years. This provides opportunities to add new hardware and functions into the next generation smartphones.

Our design and implementation of AutoCharge can be further improved. For example, our algorithm on smartphone detection may be improved in following aspects. First, besides the basic feature of rectangle shape, one may use more features such as buttons at certain positions to reduce false positive. Second, once a detected rectangle is confirmed as a smartphone, the algorithm may remember the shape of the smartphone and use it for detecting the same smartphone in the next time. This may be particularly helpful when a user uses the same light charger to charge the same smartphone in a routine matter, which is a common case for many users. Third, one may consider supporting smartphones which are not in a rectangle shape. Furthermore, it is also possible not using any predefined shape at all. Instead, the algorithm automatically learns the shapes of smartphones using various machine learning techniques. We plan to work on these optimizations.

As the main goal of this paper is to demonstrate the feasibility of automatic smartphone detection, our prototype implementation is far away from a commercial product. To become a real product for users to use in practical settings, our prototype must be improved in many aspects. For example, we must use a specially designed light to replace the T6 LED Flashlight; we must replace the Kinect sensor using a dedicated camera; we must not depend on a separate PC to run the software; and we must tightly integrate all the components together. We leave these improvements as future work.

8. Related Work

There have been efforts to solve the pain of smartphone charging. For example, some smartphones provide special versions with a large battery to reduce the times of charging [20, 21]. Doing so increases the weight of those smartphones and conflicts with the design trend of making smartphones lightweight. Some users use an external battery pack (sometimes called as a power bank [19]) to power their smartphone on-the-go. However, the users need to carry a separate device and the power bank device itself also needs to be charged. Wireless charging pads [15, 16] make smartphone charging easier because the users do not need to plug in a wire. However, the users still need to explicitly put a smartphone onto a small wireless charging pad, resulting in the similar user experience as wired chargers. Solar charging is promising for transparent smartphone charging [1, 5, 6]. However, it only works in outdoor spaces and heavily depends on time and weather conditions.

Our AutoCharge approach builds on top of existing solar charging technique but uses it in indoor spaces. To do it, we design a light charger to generate a proper light beam to charge a smartphone. Furthermore, different from existing smartphone charging approaches, we design a smartphone detection and tracking system which is able to automatically locate a smartphone on a desk and charge it without any explicit effort from the user. As a result, AutoCharge is able to reduce the burden of smartphone charging and improve the user experience.

Wireless power is also promising for remote power transfer [7, 8, 9]. However, due to safety issues caused by electromagnetic radiation, wireless power is mainly used in very short range [15, 16], in very low power [10] or in other extreme scenarios such as in outer space or for military purposes [12]. Our AutoCharge approach can also leverage (i.e., replacing the light charging part) wireless power once the techniques become mature and safe enough to be used in our targeted usage scenarios.

Using large wireless charging pads may enable automatic smartphone charging. For example, if we can build a wireless charging pad which is large enough to cover a whole table, then whenever a smartphone is put on the table it can be automatically charged without any explicit effort from the user. This will provide the same user experience as what our AutoCharge approach can do. However, this approach has some disadvantages. First, although there is on-going research work in reducing the cost of wireless charging pads [17, 18], large wireless charging pads may be too expensive for end-users, given that a small Nexus 4 wireless charger costs $59.99 [16]. Second, it may need to rebuild a table to integrate a large wireless charging pad into it. Third, such a large wireless charging pad is hard to move to be used in other places. Our AutoCharge approach does not require any changes to existing tables and a light charger is easy to move from one table to another.

Besides improving smartphone charging, the power problem may also be addressed from the opposite direction: reducing the power consumption of smartphones. Much research has been done in this direction, including improving the operating system power management [26], reducing the power consumption of hardware components [22, 24] and optimizing the power performance of applications [23, 25]. Our work is complementary with them.

9. Conclusions

In this paper we proposed and designed AutoCharge, a new approach that enables automatic smartphone charging. The key idea of the AutoCharge approach is identifying the opportunities of smartphone charging from a user's existing action of putting a smartphone on a desk and automatically charging the smartphone without requiring explicit effort from the user. To achieve it, we first leverage mature solar charging technique but use it in indoor spaces. We design a dedicated light charger to generate a light beam to charge a smartphone without a wire and address the practical issues of indoor light charging. Then, we employ a camera-based system to automatically detect and track smartphones on a desk. We develop a fast image processing algorithm which identifies smartphones from pictures. Once a smartphone is detected, we further use a rotator to track the smartphone and project a light beam onto it to charge it. The whole process is totally automatic and transparent from the user. As a result, our AutoCharge approach significantly reduces users' burden in smartphone charging and improves the user experience. The camera system can also decide the battery status of the smartphone for on demand charging and detect obstacles for safe charging.

We have implemented a prototype system of AutoCharge. Experimental results show that our prototype implementation is able to quickly detect a smartphone on a desk in various settings and charge it as fast as existing wired chargers. Despite that our prototype implementation is still far away from a real product and may be further improved in many aspects, we have demonstrated the feasibility and made a significant step towards automatic smartphone (and other mobile devices) charging.

References

[1] Wysips, An energy revolution, http://www.wysips.com/mobile-phone/performances/.

[2] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, T. Park and J. Song, Seemon: Scalable and Energy-Efficient Context Monitoring Framework for Sensor-Rich Mobile Environments, In *MobiSys*, 2008.

[3] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury and A. T. Campbell, The Jigsaw Continuous Sensing Engine for Mobile Phone Applications, In *SenSys*, 2010.

[4] S. Nath, ACE: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing, In *MobiSys*, 2012.

[5] E. Becquerel, Mémoire sur les effects électriques produits sous I'influence des rayons solaires, *Computes Rendus* 9: pp. 561-567, 1839.

[6] R. Williams, Becquerel Photovoltaic Effect in Binary Compounds, *The Journal of Chemical Physics* 32 (5): pp. 1505-1514, 1960.

[7] W. Brown, The History of Power Transmission by Radio Waves", *Microwave Theory and Techniques, IEEE Transactions on*, vol. 32, no. 9, pp. 1230-1242, Sep. 1984.

[8] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher and M. Soljacic, Wireless Power Transfer via Strongly Coupled Magnetic Resonances, *Science*, vol. 317, no. 5834, pp. 83-86, 2007.

[9] M. I. Abbasi, S. A. Adnan, M. Amin and F. Kamran, Wireless Power Transfer Using Microwaves at 2.45 GHz ISM Band, In *Proceedings of Internetional Bhurban Conference on Applied Sciences & Technology*, 2009.

[10] A. P. Sample, D. T. Meyer and J. R. Smith, Analysis, Experimental Results, and Range Adaptation of Magnetically Coupled Resonators for Wireless Power Transfer, *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 2, pp. 544-554, 2011.

[11] A. Christ, M. G. Douglas, J. M. Roman, E. B. Cooper, A P. Sample, B. H. Waters, J. R. Smith and N. Kuster, Evaluation of Wireless Resonant Power Transfer Systems with Human Electromagnetic Exposure Limits, *Electromagnetic Compatibility, IEEE Transactions on*, vol. pp, no. 99, pp. 1-10, 2012.

[12] J. L. Li, Wireless Power Transmission: State-of-the-Arts in Technologies and Potential Applications, In *Proceedings of the Asia-Pacific Microwave Conference*, 2011.

[13] R. O. Duda and P. E. Hart, Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Comm. ACM, Vol. 15, pp. 11-15*, January, 1972.

[14] Microsoft, Kinect for Xbox 360, http://www.xbox.com/en-US/kinect.

[15] Nokia Wireless Charging Plate, http://www.nokia.com/us-en/products/accessory/dt-900/.

[16] Nexus 4 Wireless Charger, https://play.google.com.

[17] A. Noda and H. Shinoda, Selective Wireless Power Transmission Through High-Q Flat Waveguide-Ring Resonator on 2-D Waveguide Sheet, *Microwave Theory and Techniques, IEEE Transactions on*, vol. 59, no. 8, 2011.

[18] Shinoda Lab, Two-Dimensional Signal/Power Transmission Technology for In-Room Networking and Circuit Integration in Fabrics, http://www.alab.t.u-tokyo.ac.jp/~shino/2Dcom/2DST.html.

[19] EasyAcc Power Bank, http://www.amazon.com.

[20] Huawei Ascend D2 smartphone, http://www.huaweidevice.com.

[21] XiaoMi smartphone, http://www.xiaomi.com/mi2.

[22] H. Han, Y. Liu, G. Shen, Y. Zhang and Q. Li. DozyAP: Power-Efficient Wi-Fi Tethering. In *MobiSys*, 2012.

[23] R. Mittal, A. Kansal, and R. Chandra, Empowering developers to estimate app energy consumption. In *MobiCom*, 2012.

[24] M. Dong and L. Zhong, Chameleon: a color-adaptive web browser for mobile oled displays. In *MobiSys*, 2011.

[25] M. Ra, B. Priyantha, A. Kansal, and J. Liu, Improving Energy Efficiency of Personal Sensing Applications with Heterogeneous Multi-Processors. In *UbiComp*, 2012.

[26] L. Zhong, Energy-efficient mobile system design: The User's perspective, Ph.D. dissertation, Dept. of Electrical Engineering, Princeton University, 2005.

[27] Texas Instruments, MSP430 Microcontroller, http://www.ti.com.

[28] S. Suzuki, Topological Structural Analysis of Digitized Binary Images by Border Following, Computer Vision, Graphics, and Image Processing, vol. 30, pp. 32-46, 1985.

[29] J. Sklansky, Finding the Convex Hull of a Simple Polygon, *Pattern Recognition Letters*, vol. 1, no. 2, pp. 79-83, 1982.

[30] UltraFire CREE XM-L T6 Focusing LED Flashlight torch, search http://www.amazon.com.

[31] DFRobot FIT0046 DF15MG Tilt/Pan Kit, search http://www.dfrobot.com.

[32] OpenNI, http://www.openni.org.

[33] OpenCV, http://opencv.org.