

SGASDP: Smart Glasses Application Software Development Platform

Jing Chen

Department of Electrical Engineering
National Cheng Kung University
Tainan City, Taiwan ROC
e-mail: jchen@mail.ncku.edu.tw

Yu-Chieh Pai* Jian-Hong Liu

Institute of Computer and Communication Engineering
National Cheng Kung University
Tainan City, Taiwan ROC
e-mail: {q36031203*, liuken}@rtpc06.ee.ncku.edu.tw

Abstract—Smart glasses have inspired a variety of applications with innovation in many areas ranging from personal life to industry. However, owing to some uncertainties in application trends, which raise the issues concerning security and privacy, and the difference in the human-computer interaction between smart glasses and popular smart phones, not only consumers hesitate, but also the makers do not seem to invest much in supporting application software development. Therefore, the developers of smart glasses applications mostly could only rely on traditional Google Android application development tools and suffer from cost in terms of work load and time. This paper presents an application software development platform for smart glasses, namely SGASDP (Smart Glasses Application Software Development Platform), which is intended to increase the productivity and the efficiency in building smart glasses applications, as well as to help enhance the user experience of developers during the development process. The development of SGASDP adopts as its base the original application development platform of Google Android systems. This work extends the part of software development kit (SDK) by integrating, as built-in components, an SDK for live streaming and an SDK for voice recognition. In addition, development aids are added into the platform, which include a smart glasses display layout design tool and an expandable API usage helper. SGAPDP has been tested via developing application software for products of two local smart glasses makers. The success in the development demonstrates the contribution of this work.

Keywords- smart glasses; wearable computing; application development; Android; application launcher

I. INTRODUCTION

Smart glasses or smartglasses is a wearable computer equipped with a near-eye display or optical head-mounted display (OHMD) [1]. It is considered as one of the latest innovations among wearable computing devices, which has emerged to address the limitation of mobile devices, such as requiring dedicated attention, hand-held and operating, etc. Since Google Glass was introduced, smart glasses products quickly prevailed [2][3], and many applications, not only in personal daily life, but also in many areas which range from education to industry have been inspired [4]-[8]. However, not only consumers appear to hesitate but also makers do not invest much effort in supporting development of application software. The main reasons might include the uncertainty in application trends with concerns of security and privacy, and the difference in human-computer interaction between smart

glasses and popular smart phones [9][10]. Further, Google Inc. enforced a policy that is very different from the one for Android system in developing application software [2][11]. Consequently, in developing applications for Android-based smart glasses products, the developers could only rely on the traditional Android application development tools [12] and might suffer from more work load and cost of time.

Despite that the production of Google Glass is ceased [2], applications of smart glasses in many areas, such as health care and medical field [13][14], automotive industry [15], and those in which in-field hands-off operating is required [16], can be seen quite flourishing. A software development platform that helps build smart glasses applications thus is highly desirable. For example, two scenarios of smart glasses applications in medical field are shown in Figure 1. In (1), smart glasses is used in an operating room for live streaming, which transmits or broadcasts the process of a surgery operation, so that other physicians can offer advices in real-time [17]; or to make it a teaching or training aid for medical courses. The scenario in (2) shows that smart glasses, with its particular features, can cast information on the lens and can provide the doctor an intuitive way to consult patient’s medical records or other information while voice recognition is used to help operate smart glasses; the doctor’s hands thus are free for other needs. The application functionality of smart glasses required in these scenarios calls for a software development platform which can help develop application software for smart glasses efficiently and conveniently.

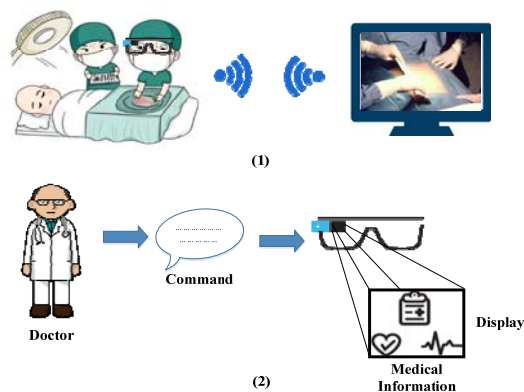


Figure 1. Applications of smart glasses

This paper presents the development of SGASDP, which is an application software development platform intended to

TABLE I. COMPARISON OF SMART GLASSES PRODUCTS

Maker	Google [2][18]	Chipsip [19]	Jorjin [20]	Sony [21]	Epson [22]	Lenovo/Vuzix [23]
PRODUCT	Google Glass	SiME Smart Glasses	Jorjin Smart Glasses Solutions	SmartEyeglass	Moverio	Lenovo Vuzix Smart Glasses
MODEL	X1	SiME	JGK-S101	SED-E1	BT-200	M100
DATE	2014/04	2015/03	N/A	2014/09	2014/12	2014/09
CPU	OMAP 4430	Newton32 SiP (Dual Cortex-A9)	OMAP 4460	N/A	OMAP 4460	OMAP 4460
RAM	1 GB	1 GB	1 GB	N/A	1 GB	1 GB
DISPLAY	Monocular	Monocular	Monocular	Binocular	Binocular	Monocular
RESOLUTION	640 × 360	800 × 480	800 × 480	419 × 138	960 × 540	400 × 240
WIRELESS	1. Wi-Fi 2. Bluetooth 4.0 3. GPS	1. Wi-Fi 2. Bluetooth 4.0 3. GPS	1. Wi-Fi 2. Bluetooth 4.0 3. GPS	1. Wi-Fi 2. Bluetooth 3.0	1. Wi-Fi 2. Bluetooth 3.0 3. GPS	1. Wi-Fi 2. Bluetooth 4.0 3. GPS
O/S	Glass OS / Android 4.4	Android 4.4.2	Android 4.2.2	Android 4.1 or later	Android 4.0.4	1. Android 4.0.4 2. Customized Android
SDK	1. Android SDK 2. GDK	1. Android SDK 2. SiME SDK	1. Android SDK 2. Jorjin SDK	1. Android SDK 2. SmartEyeglass SDK	1. Android SDK 2. MOVERIO BT-200 SDK	1. Android SDK 2. Vuzix SDK
SUPPORT	Developer Web site	Company team	Company team	1. Stackoverflow 2. Developer Web site	Developer Web site	Developer Web site
NOTES	1. Developer is Google X 2. H/W maker is Foxconn 3. Discontinued 2015/01		Jorjin provides BSP as the maker SDK	1. Developer Edition 2. H/W spec. are not all publicly shared.		Cobranded by Lenovo and Vuzix

help increase the efficiency and the productivity in building smart glasses applications. The goals of this development include the following desirable achievements on SGASDP:

- The platform would take Android-based system as the target of applications. This is because that Android not only has been very popular in mobile devices, but also is the base system of Google Glass. In addition, it can allow experienced developers to avoid the learning curve in adapting to a new development platform.
- The platform should support the common features and requirements of smart glasses user interfaces, such as horizontal display and voice control.
- The platform should support functionalities commonly required in building various smart glasses applications, such as the flexibility in integrating suitable SDK.
- The platform would, through integrating useful and handy tools or development aids, provide convenience and help enhance the user experience of developers in the development process.

In addition, this platform would be targeted on developing user space application software. By adopting Android system to be its base, the impact from system level dependency and hardware variety would not cause much difficulty.

The rest of this paper is organized as follows. Section II discusses briefly the background and related works. Section III presents the development of SGASPD, from the design to implementation while Section IV discusses and evaluates the current implementation. Section V concludes this paper.

II. BACKGROUND AND RELATED WORKS

This section gives a summary on developing application software for smart glasses products, which forms the basis of this work.

Smart glasses and portable devices (e.g., smart phones and tablet computers) have similar hardware components, such as processor, memory and sensor modules. The main

differences, however, are the operating methods and the horizontal display which relate to user interfaces (UI) or human-computer interaction. To cope with these differences, Google Glass comes with new UI called timeline cards [24] and Glass Development Kit (GDK) [18]. The timeline is the main user interface that is composed of 640×360 pixel cards. Users can scroll through different sections of the timeline so as to reveal cards in the past, present, and future. Figure 2 is Google Now weather card which shows relevant information of current weather [24]. GDK is an add-on to Android SDK and supports the functions of voice control, gesture detector, cards, and others. GDK makes existing Android SDK work on Google Glass and allows developers to build applications that directly run on Google Glass. However, GDK is only available for Google Glass and those applications developed using GDK can only run on Google Glass [11].

Due to space limit and to in order to focus on the main theme of this paper, the details of smart glasses products cannot be included here. Table I presents a brief comparison on Google Glass and smart glasses products of other known makers. This comparison is by no means exhaustive but it can be observed from the table that Android system is quite popular. Therefore, the software developers of smart glasses applications in general can leverage the development tools and environment of Android SDK.



Figure 2. Example of Google Live Card [24]

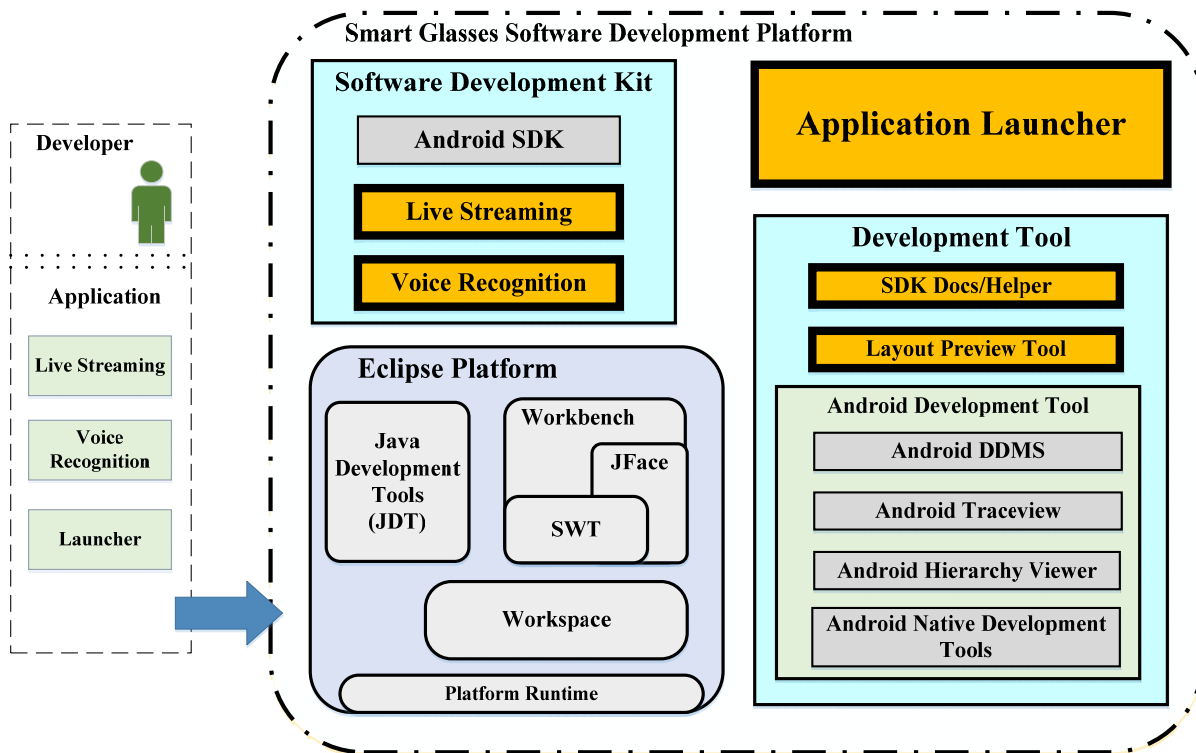


Figure 3. The architecture overview of Smart Glasses Application Software Development Platform (SGASDP)

As shown in Table I, the makers of smart glasses provide their proprietary SDK, as a supplement to Android SDK, to help access maker-specific devices as well as enable the third party application development. This, mostly, is not the same case as the GDK of Google Glass, because these maker SDK do not, if any, support all the functions provided in GDK. An example is the UI of timeline and cards, as mentioned earlier. Developing applications for these smart glasses products thus would have to handle extra work load in addition to coping with application logic.

III. THE DEVELOPMENT

This section describes the development of SGASPD, which includes the design, its architecture and its implementation. SGASPD was conceived with the application development platform of Android systems in mind. The reason is not only that Android system has a large, and is still growing, market share in the arena of mobile platforms, but also it is relatively stable such that hardware variety in smart glasses products would not impose impact upon developing application software. In addition, for experienced Android software developer, the burden of learning and becoming used to a new development platform or environment can be greatly reduced. Of course, the reasons of open source and free availability both have shares in making the decision.

A. The Design

Figure 3 depicts the architecture overview of SGASPD and shows the main design consideration. The dotted boxes at the left-hand side in the figure shows some functionalities that might be commonly desirable in applications of smart

glasses, such as live streaming and voice recognition. These functional requirements and other potential ones, such as gesture-based control, might suggest that a mechanism of this platform to support implementing functionality specific to smart glasses applications is helpful and desirable. SDK would be suitable to serve as package of functionality while an SDK manager is designed to harness the flexibility in adding and removing SDK.

The launcher box below the box of voice recognition in Figure 3 stands for a customized application launcher which is intended to replace the native Android Launcher (also known as Android launcher app.) because the user interfaces of smart glasses are very different from those of Android smart phones or tablet devices. A new application launcher is thus necessary and therefore it is included in the design.

B. The Architecture

Based on the design described above, the big chain block in Figure 3 shows the main architecture of SGASPD, which is composed of four main parts:

- **Software Development Kits:** These components include not only the SDK required to develop Android apps [25], but also the SDKs pertinent to implementing desirable functionality of smart glasses applications.
- **Eclipse Platform:** Eclipse [26] is adopted as the base of SGASPD. The components are Java Development Tools, Workbench, Workspace, and Platform Runtime.
- **Application Launcher:** This a light weight UI included specifically for smart glasses applications. It takes as reference the concept of “timeline cards” [24] from GDK to fit the horizontal view of smart glasses.

- **Development Tools:** These components are intended to provide handy assistance to developers, which include a design aid for the display layout of application, preview utility, usage guide and other Android development tools. With the built-in capability of Eclipse [27], components can be added or removed as needed. SGASDP therefore features expandability as well as flexibility.

C. The Implementation

SGASDP is implemented taking as its base the original application development platform for Android system [25], of which the core components are Eclipse platform and Android SDK. SGASDP extends the part of SDK with an SDK for live streaming and an SDK for voice recognition. The part of development aids is realized by porting an open source display layout design tool, namely DroidDraw [28], and implementing an Eclipse plug-in module to serve the role of the SDK Docs/Helper [27]. The application software launcher, namely SGASDP Launcher, is built as a user-space program which supports landscape view of smart glasses and can be configured to replace the native Android launcher. All the implementation work was done on an Intel x86 computer running Windows 7 operating system. The components were created using Java programming language.

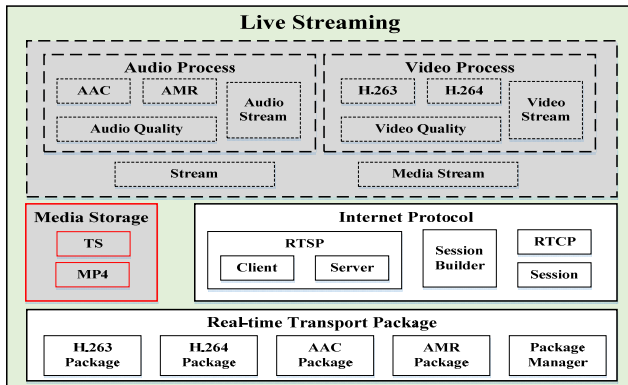


Figure 4. The composition of Live Streaming SDK

(1) **Live Streaming SDK:** This SDK is realized through porting and rebuilding the libstreaming package [29]. Figure 4 shows the structure of Live Streaming SDK. It can stream data from the camera or microphone of an Android device using Real-time Transport Protocol (RTP) [30] over User Datagram Protocol (UDP) [31] and can achieve 1080p. It has five parts, namely Audio Process, Video Process, Real-time Transport Package, Internet Protocol, and Media Storage.

(2) **Voice Recognition SDK:** This SDK is implemented by rebuilding the PocketSphinx package of Sphinx project, an open source project to build toolkit for speech recognition [32][33]. The rebuilding work includes compiling the Linux version of the package into the format of dynamic linking library, and then building into Android SDK using Android NDK [34]. This SDK supports off-line voice recognition and developer can customize it with different dictionary files to be integrated into the application.

(3) **SGASDP Launcher:** This launcher is implemented to fit the horizontal display layout for smart glasses and can be

used to replace the native Android launcher app. It provides a horizontal list view of icons and supports voice-operating capability to handle application triggering behaviors from the user. This launcher should be configured into Android App Manifest [35] of the target system in order to make it the first program to run when target Android system starts. Figure 5 shows an example display of this SGASDP Launcher.

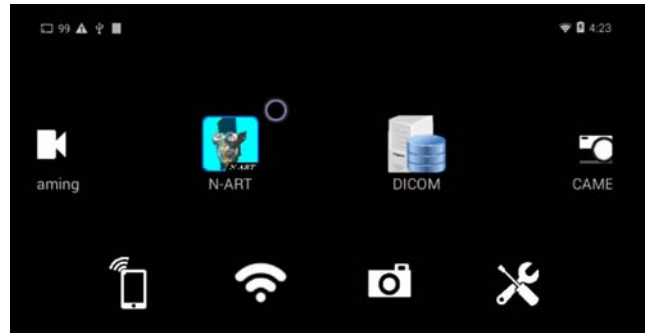


Figure 5. The landscape view of SGASDP Launcher display

(4) **SDK Docs/Helper:** The function of SDK Docs/Helper is to help developers in using SGASDP and the integrated SDKs. In order to be easily integrated as well as managed in SGASDP, it is implemented using the Plug-in Development Environment (PDE) [27] and the Standard Widget Toolkit (SWT) [36] of Eclipse.

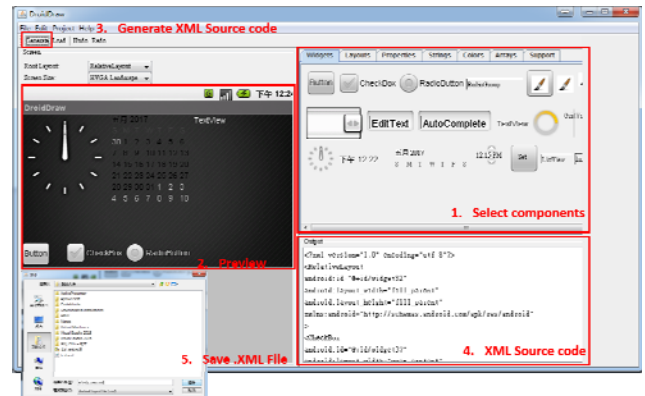


Figure 6. The GUI of the layout design aid and preview tools of SGASDP

TABLE II. RESOLUTIONS SUPPORTED BY THE LAYOUT PREVIEW TOOL

Standard	Resolution	Aspect Ratio
QVGA	320 × 240	4:3
HVGA	480 × 320	3:2
nHD	640 × 360	16:9
VGA	640 × 480	4:3
SVGA	800 × 600	4:3
WVGA	800 × 480	5:3
FWVGA	854 × 480	~16:9
qHD	960 × 540	16:9

(5) **Display layout design tool:** The implementation of this tool adopts and extends DroidDraw open source project [28]. DroidDraw is intended to be a graphical user interface (GUI) builder for the Android platform, which can help developers to design the display layout of their applications.

An example is shown in Figure 6. In addition, as listed in Table II, more display resolutions are implemented to cope with product variety of smart glasses. The functions of this tool help design including preview a display layout, and convert a layout design to source code in XML format to be imported to the target application.

IV. EVALUATIONS

To demonstrate the functionality and the effectiveness, the current implementation of SGASDP was used to develop example application software for the scenarios mentioned in Section I. The developed software features real-time live streaming using Wi-Fi connection, as shown in Figure 7, and voice control via the SGASDP Launcher. It runs well on the smart glasses products of two local makers, namely ChipSip [19] and Jorjin [20]. The smart glasses from ChipSip, namely SiME Smart Glasses, is equipped with Newton 32 SiP of Dual Cortex-A9 processors [19], while that from Jorjin, Jorjin Smart Glasses, is equipped with OMAP4460 processor [20]. Both are operating at the clock rate of 1.2 GHz. A test case of real-time live streaming on SiME smart glasses is shown in Figure 8.



Figure 7. Real-time live streaming using Wi-Fi connection



Figure 8. Testing real-time live streaming on SiME smart glasses

During the process of developing the example application software, two handy tools provided by SGASDP, namely the layout design tool and SDK Docs/Helper, are used as shown in Figure 6 and Figure 9 respectively. Figure 6 demonstrates the typical five-step procedure from selecting components to saving the generated XML code defining the display layout. Figure 9 is a snapshot of screen display during the software development and the window located in the lower part of the main view is the SDK Docs/Helper UI which demonstrates the functionality of querying API usage.

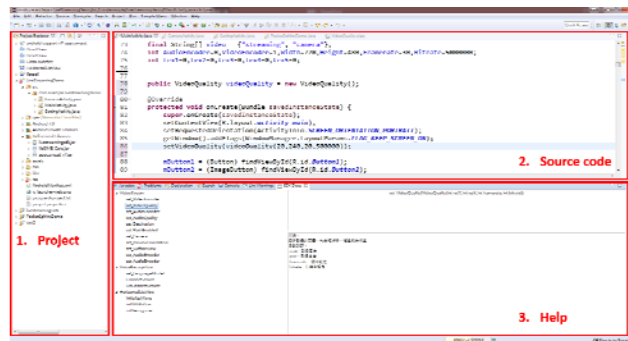


Figure 9. Using SDK Docs/Helper

```

root@nt32_araile:/ # logcat -d -b events | grep "boot"
I/boot_progress_start( 106): 12756
I/boot_progress_preload_start( 106): 14639
I/boot_progress_preload_end( 106): 20200
I/boot_progress_system_run( 439): 20524
I/boot_progress_pms_start( 439): 20734
I/boot_progress_pms_system_scan_start( 439): 21140
I/boot_progress_pms_data_scan_start( 439): 25661
I/boot_progress_pms_scan_end( 439): 32274
I/boot_progress_pms_ready( 439): 32428
I/boot_progress_ams_ready( 439): 34702
I/boot_progress_enable_screen( 439): 42729
    
```

Figure 10. Using Android ADB to retrieve log data

TABLE III. COMPARISON OF SMART GLASSES LAUNCHERS

Launcher	Startup Speed	Operating	Display
Native Launcher	Slow	Touch	Portrait
SGASDP Launcher	Fast	Touch & voice	Landscape

TABLE IV. COMPARISON OF LAUNCHER STARTUP TIME

Launcher	Average	Max.	Min.	Std. dev.
Native Launcher	44.463	47.352	42.102	1.760
SGASDP Launcher	43.712	46.145	42.132	1.283

Unit: second

For evaluating application launcher of SGASDP, listed in Table III are the characteristics of two launchers in which the Native Launcher is the default Android application launcher that is built in the Android system of SiME Smart Glasses. Table IV compares the performance in terms of startup time from cold-start state, which can be obtained via measuring the booting time as the typical launching activity is included in the cold-start booting process. So the booting time can be calculated from the time interval between the timestamps of “boot_progress_start” and “boot_progress_enable_screen” recorded in system log file. The results listed in Table IV are obtained from booting each launcher 10 times and retrieved the timestamps using Android Debug Bridge (ADB) [37] as shown in Figure 10. It is seen that the SGASDP Launcher with voice control capability incurs no significant loading.

V. CONCLUSIONS

This paper presents SGASDP, smart glasses application software development platform, which is aimed to increase

the productivity as well as the efficiency in building smart glasses applications. SGASDP features the following main beneficial characteristics: (1) live streaming SDK which helps implement the functions of video or audio streaming in applications; (2) customizable voice recognition SDK which supports off-line voice recognition capability; (3) application launcher which provides user interface of horizontal display and voice-control operating functions; (4) a display layout design tool which helps design and preview the layout of application display. Besides, SGASDP can be transplanted easily through directly copying the relevant files to the new environment when needed, and developers can customize or enhance its capability by integrating their own development kits or tools to make this platform more fit and powerful.

Test results demonstrate that SGASDP works properly and there is no burden, for developers with little experience, in learning to operate this platform. The main contribution of this work is a software development platform that effectively helps developers in building smart glasses applications.

ACKNOWLEDGMENT

This work is partially sponsored by Ministry of Science and Technology (MOST) of Republic of China (ROC) under the contract MOST106-3011-E-006-001.

REFERENCES

- [1] Wikipedia contributors, "Smartglasses", <http://en.wikipedia.org/wiki/Smartglasses>, accessed on 2017-07-28.
- [2] Wikipedia contributors, "Google Glass", http://en.wikipedia.org/wiki/Google_Glass, accessed on 2017-08-16.
- [3] P. Lamkin, "The best smartglasses 2017: Snap, Vuzix, ODG, Sony & more", July 10, 2017, <https://www.wearable.com/headgear/the-best-smartglasses-google-glass-and-the-rest>, accessed on 2017-07-30.
- [4] Google Glass, "Google Glass: What Is It and How Can It Change Our Lives", CreateSpace Independent Publishing Platform, March 9, 2014, ISBN 978-1497301719.
- [5] E. Butow and R. Stepisnik, "Google Glass for Dummies", For Dummies, 1st Edition, April 21, 2014, ISBN 978-1118825228.
- [6] M. Wells, "How Google Glass Can Be used in Education", <http://www.gettingsmart.com/2014/07/google-glass-can-used-education/>, July 19, 2014, accessed on 2017-07-24.
- [7] M. C. O'Connor, "Smart Glasses Finding Work Across Industries", Jan. 6, 2015, <http://www.iotjournal.com/articles/view?12576>, accessed on 2017-08-05.
- [8] M. Margolis, "Smartglasses: Industrial Applications And Now In Retail", July 2, 2015, <https://seekingalpha.com/article/3299785-smartglasses-industrial-applications-and-now-in-retail>, accessed on 2017-08-05.
- [9] A. Oreskovic, S. McBride, and M. Nayak, "Google Glass future clouded as some early believers lose faith", Nov. 14, 2014, <http://www.reuters.com/article/us-google-glass-insight-idUSKCN01Y18E20141114>, accessed 2017-07-26.
- [10] M. Swider, "Google Glass Review", Feb. 21, 2017, <http://www.techradar.com/reviews/gadgets/google-glass-1152283/review>, accessed on 2017-07-30.
- [11] Google Inc., "Google Developer Policies", <https://developers.google.com/glass/policies>, accessed on 2017-07-01.
- [12] Android Developers, "Android Studio," <https://developer.android.com/studio/index.html>, accessed on 2017-08-01.
- [13] N. Wrzesińska, "The use of smartglasses in healthcare - review", Medtube Science, Vol. III (4), Dec. 2015.
- [14] E. Bostancı, "Medical wearable technologies: Applications, Problems and Solutions", 2015 Medical Technologies National Conference (TIPTEKNO), Bodrum, 2015, pp. 1-4.
- [15] M. Neuner, "How can the car industry benefit from Wearables", August 16, 2016, <https://www.wearable-technologies.com/2016/08/how-can-the-car-industry-benefit-from-wearables/>, accessed on 2017-08-13.
- [16] J. Dutschke, "Enhancing the Maintenance Vision", <http://www.mromagazine.com/features/enhancing-the-maintenance-vision/>, accessed on 2017-08-18.
- [17] D. F. Carr, "Google Glass Enables Surgeons To Consult Remotely", Nov. 29, 2013, <http://www.informationweek.com/healthcare/mobile-and-wireless/google-glass-enables-surgeons-to-consult-remotely/d-d-id/1112837?>, accessed on 2017-08-15.
- [18] Glass Explorer Edition, "Glass Development Kit (GDK)", <https://developers.google.com/glass/develop/gdk/>, accessed on 2017-05-31.
- [19] ChipSip Technology Co., "SiME Smart Glasses", <http://www.chipsip.com/computing/index.php?mode=data&id=126>, accessed on 2017-08-17.
- [20] Jorjin Technologies Inc., "Jorjin Smart Glasses Solution", http://www.jorjin.com/solution_content.php?id=28, accessed on 2017-07-26.
- [21] Sony Developer World, "Develop SmartEyeGlass App", <https://developer.sony.com/develop/wearables/smarteyeglass-sdk/>, accessed on 2017-08-18.
- [22] Epson Inc., "Epson Moverio BT-200", https://www.epson.com/Support/Wearables/Moverio/Epson-Moverio-BT-200/s/SPT_V11H560020, accessed on 2017-07-31.
- [23] Vuzix Inc., "M100 Smart Glasses", <https://www.vuzix.com/Products/m100-smart-glasses>, accessed on 2017-08-15.
- [24] Glass Explorer Edition, "User Interface", <https://developer.google.com/glass/design/ui>, accessed on 2017-08-01.
- [25] Wikipedia contributors, "Android Software Development", http://en.wikipedia.org/wiki/Android_software_development, accessed on 2017-08-13.
- [26] Eclipse, "Eclipse Platform Technical Overview", <http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>, accessed on 2017-08-17.
- [27] Eclipse, "PDE", <http://www.eclipse.org/pde/>, accessed on 2017-08-15.
- [28] L. Meyer, "DroidDraw", <https://github.com/sosieuxme/DroidDraw>, accessed on 2017-08-16.
- [29] Simon (fyhertz), "libstreaming", <https://github.com/fyhertz/libstreaming>, accessed on 2017-08-02.
- [30] "RTP: A Transport Protocol for Real-Time Applications", <https://tools.ietf.org/html/rfc3550>, accessed on 2017-08-15.
- [31] "UDP: User Datagram Protocol", <https://tools.ietf.org/html/rfc768>, accessed on 2017-08-15.
- [32] CMUSphinx, "Open Source Speech Recognition Toolkit", <https://cmusphinx.github.io>, accessed on 2017-08-19.
- [33] CMUSphinx, "Overview of CMUSphinx toolkit", <http://cmusphinx.sourceforge.net/wiki/tutorialoverview>, accessed on 2017-08-12.
- [34] Android Developers, "Android NDK", <https://developer.android.com/ndk/index.html>, accessed on 2017-08-09.
- [35] Android Developers, "App Manifest", <https://developer.android.com/guide/topics/manifest/manifest-intro.html>, accessed on 2017-08-09.
- [36] Eclipse, "SWT: The Standard Widget Toolkit", <https://www.eclipse.org/swt/>, accessed on 2017-08-15.
- [37] Android Studio, "Android Debug Bridge (adb)", <https://developer.android.com/studio/command-line/adb.html>, accessed on 2017-08-01.