other 7 participants did not indicate explicitly whether or not they used a Braille display.

We found that many participants encountered issues with IDEs. The accessibility challenges ranged from completely unusable to advanced features proving inaccessible. All but three respondents to the survey had accessibility challenges with the IDEs that they used most often. Common problems indicated on the survey were interface builders (6 participants), debuggers (3 participants), syntax highlighting (2 participants), and diagrams (2 participants).

Additionally, participants clarified that while an IDE may be accessible, learning to use it was much more complex. Most guides for IDEs that participants attempted to use were geared toward sighted users, often littered with directions like "click here" which made them irrelevant to a blind programmer. Determining keyboard equivalents was complex, often leading participants away from assignments in search alternatives. Unsurprisingly, we learned that participants like i9 with prior knowledge of the IDEs found it essential for setting them up to succeed.

Attempting to, and failing to, use inaccessible IDEs deterred some participants from trying new ones. One participant said:

> *And like my confidence was pretty low that these programs would even work if I had the time to spend. So I was kind of going under this assumption that they wouldn't anyway. And I think I would have been right in most cases, but I don't know. ... So I didn't have time to spend and you know, put like 40 hours into Eclipse and then learn, oh ok cool, you know, it's not accessible. - i1*

The burden of spending so much extra time to no avail stuck with i1 to the point that potentially wrongfully assuming an IDE would be inaccessible was a more efficient use of his time.

For some of our participants like i8, choosing to avoid learning new IDEs during university was a purposeful choice that, in the short term, mitigated some of the overhead skill acquisition in order to keep up with his classes. However, such choices came with tradeoffs. I8 is now learning how to use a debugger as he never used one in college. He learned that his previous strategies were ineffective in comparison.

Another barrier came when participants were asked to complete assignments with unique software downloaded onto specific machines. Often, this software was built for beginner students meant to direct their attention from IT to learning programming concepts. But these computers remained restrictive disallowing some of our participants from downloading important assistive technologies onto them.:

> *But the problem was that you could only access that server from the lab machines and the lab machines didn't have any speech on them. So I couldn't use it all. So I actually – I ended up having to set up my own machine and you know set-up Oracle, set up PHP, Apache, and all things like that. But I had to get the assignment in, you know, to the same deadline that everyone else did. – i6*

Participants were burdened when they had to configure separate machinery. This often required extra work which distracted them from coursework. Additionally, lab spaces serve additional purposes other than just access to the technology. Lab spaces can serve as a communal space for students and without access to the machines in these spaces may be less inviting to blind students.

4.2 Accommodations

One of the factors that affected a student's experience was the quality of the accommodations that were provided. In this section, we will discuss their experiences related to these accommodations in regard to accessing lectures, materials, and assignments.

4.2.1 Lecture. One of the major problems blind students face in lecture is missing context, as what is spoken does not replicates visual content:

> *The professor would say something like 'sizeof int,' right? So he would leave out that it was actually 'sizeof left parenthesis int right parenthesis semi-colon. ...' So you miss all that. So when I first started learning C in college, I would actually write 'sizeof space int semi-colon' on my exams. –i2*

Issues such as this were compounded when the lecturer handwrote and gestured to content rather than speaking it out. I2 acknowledged that requesting content be read out loud presented a new experience for the lecturer and would sit at the front of the room as a physical reminder of the accommodation. When the professor did not verbalize their handwriting and did not provide the same information in accessible course materials, participants relied on friends in the class whose notes were likely also inaccessible; for i1, this meant receiving unreadable notes in MathType or Microsoft Word's Equation Editor.

4.2.2 Materials. Our participants had difficulty obtaining course materials such as textbooks and handouts in an accessible format on time. Participants tended to prefer one of a few different mediums: Braille, audio, or electronic but each format had its pros and cons.

Producing Braille textbooks is time consuming, requires specialized equipment and expertise, and is expensive to create. Those who received Braille textbooks cited large delays interfering with their learning. I2 ordered braille transcription 2 months in advance to receive them when classes started. I1 would sometimes not get the braille book until over half way through the term, if at all. In order to have access to the book for the entire term, he would sometimes drop and retake a course. To mitigate the cost and time factors, i4 ordered books from a Braille library. However, they often were not the same book assigned, but the faculty would look them over and determine if they adequately covered the topics.

Multiple (i2, i4, i7, i8) participants used audiobooks from sites like Learning Ally [21]. Books from Learning Ally were appreciated because they recruited readers versed in the subject.

This had two benefits. The first was that they could read the subject material in a way that made sense. For things like code, they would know what needed to be read, such as punctuation. Additionally, they would often describe the diagrams by prioritizing the pertinent information. But they also had a downside. One participant found the spoken code very hard to listen too:

> *So, books and audio when it comes to programming is very inefficient. Some of the readers, what they would do, is they would actually spell the entire code. So they would say f-u-n-c-t-i-o-n. Space... - i2*

Since there is no established standard for verbalizing code, audio book readers sometimes omitted information. For example, participants indicated readers would fail to communicate indentations which can be consequential for certain programming language. Additionally, it is not possible to search audiobooks, which can make them less useful for looking up information.

Participants loved obtaining an electronic format of the textbook as they were versatile and searchable. For example, participants could read content with a screen reader or refreshable braille display, and write notes and highlight important content. However, electronic formats were difficult to obtain. Three of the interview participants (i1, i5, i6) contacted the textbook publishers directly to obtain a premade electronic version, but only one actually received a textbook. Those unable to get the electronic version directly from the publisher scanned the hardcopy book and used Optical Character Recognition (OCR) software to extract text. However, OCR works best with prose and stem-related content including computer code is often interpreted incorrectly. Since correct syntax is imperative in computer science, even intermittent errors could confuse students as to whether they did not understand the material or whether their OCR version of the textbook was inaccurate. One participant's office providing accommodations for students with disabilities hired workers with knowledge of computer science and were therefore able to correct OCR-produced errors. However, if the student was unable to get the OCR corrected, explicitly mentioned by (i3, i5, i6), then they received the textbook with the afore-mentioned errors. One participant described the effect as:

> *Then scan was always the last scenario because that would have a drastic effect on my mark. You know, to the point where I was ... getting First [highest grade] for the things where I had electronic copies of books readily available to me, but I was barely passing modules where I had to get the book scanned. - i6*

Textbook formats varied but all had pros and cons. Students experimented with different types but never completely succeeded in obtaining their course materials on time in an accessible format.

Another common resource students needed access to were diagrams. In general, they were either provided in a tactile form or a verbal/textual description. Sometimes students would only get a small subset of the diagrams made accessible because conversion took time, special equipment in the case of tactile graphics, and expertise. Additionally, the source of descriptions varied. For one student, the professor included descriptions on his slides. Others relied on friends or family members to describe the images. We discovered that who provided the descriptions largely impacted their accuracy. Our participants reiterated that diagram describers need to be knowledgeable of the material:

> *And it's kind of like well, ok, I need to know where the switches are and stuff like that. ... But, you know, if you don't know about computers, you can't just look at some of these diagrams and give an accurate description of them. - i6*

Participants encountered similar miscommunication when someone served as their go-between during exams. A common scenario consisted of a reader describing questions on the exam and transcribing answers as participants dictated. Multiple participants lamented these experiences negatively. i4 described one of his tests:

> *And the problem was the person they got to actually do the writing for me was a chemistry post graduate - had never done this part of mathematics before. So he didn't know what the hell he was writing. - i4*

Timely and accessible textbooks and diagrams proved important but rarely-available components to participant success. It was important that description authors were aware of the field Second, since participants were still learning the content, they were unqualified to provide the reader instructions on what and how to communicate. However, services like Learning Ally which recruited subject matter experts to read computer science textbooks did not mitigate this barrier given the limits of audio books on searching and inconsistent descriptions given no universal reading standard. Finally, exams represented negative experiences for our participants when nonexperts not only could not describe content but when they could not transcribe answers as intended.

4.2.3 Assignments. Some participants were provided alternate assignments or exemptions during their computer science degree programs. These examples spanned using different technologies to complete coursework, building a different system, leveraging human assistance, receiving grades based on the accessible components, and being excused altogether.

One of the most common issues occurred when assignments steered students toward an inaccessible IDE. This was often not consequential as assignment requirements could be fulfilled regardless of the IDE, but it did mean that they undertook extra work to learn the IDE on their own and they may not have had access to the same tools (i.e. debugging).

Another common alteration was changing the form factor of the assignment, resulting in different output with similar application of programming concepts. Participants sometimes wrote text descriptions of non-textual answers (i5 and i9) or doing a console based application instead of GUI application (i3, i6, i10).

However, other participant scenarios exemplified two very different assignments. I9 built a chat application once when the rest of his class built a visual game. This application *"covered some common concepts and others which the game did not. So it was equal but not the same."*

The second major assignment change occurred when the participant was assigned a sighted partner to relay visual information, or when a reader completed visual aspects under the participant's direction. I6 was asked to create a drawing application. He had a sighted peer help him by watching him run his code to communicate if it succeeded, but he wasn't very motivated by the assignment:

> *And to be completely honest, I didn't put a huge amount of effort into that assignment. Because I just hated the idea of kind of spending hours and hours of making something I'd just never be able to use. –i6*

The student's motivation was decreased by having an assignment that he did not find exciting as he would be unable to actually use (and debug) the output. The low effort meant that he later realized that he did not know those topics as well, which is a challenge when topics build upon those earlier topics they did not learn well.

Some professors did not adapt assignments but ambiguously communicated they would consider the extra challenges when grading. I2, described just one of a few similar experiences:

> *He said, you know, as long as your code is written to the best of your ability. ... So he said double check the logic and stuff and you know, we will realize that you were never actually able to compile and run this program. – i2*

I2 was okay with this outcome as he felt confident about the content. However, it likely is not a good option in general. The process of understanding errors and figuring out how to fix them was part of the assignment that he was not able to do and thus he may not have learned as much as if he had been able to read the output and properly debug errors.

Finally, although rare, some participants were excused from individual assignments, portions of courses, and courses altogether. Participants cited professor disbelief in a blind student. For I6, this meant he was talked out of taking an elective. Most exempt courses consisted primarily of graphical or electronics content. Participants had to advocate strongly to enroll when the professor or department would have waived the requirement. I1 took an independent study version of a course in lieu of the established version which was inaccessible, but his attempts to adapt course content with the instructor were unsuccessful, defeating the purpose of the specialized course opportunity. Several participants relayed faculty were forthcoming about not knowing how to teach them, imagining exemption from visually-based content and assignments over constructing accessible alternatives. Common in these stories is that participants were not interested in missing out on any assignments or courses. Instead, they had to assert their right to complete all degree requirements.

Adapted or removed assignments altogether carried repercussions. Differences in assignments could be ambiguous as to whether they were scaffolding a similar learning experience or whether they would be evaluated objectively. Participants further had to rely on peers or readers, decreasing their engagement with the content. Finally, inaccessible assignments increased workload and interfere with concept mastery, deficiencies that some participants did not realize for years.

4.3 Social Impacts

We found in our interviews that social factors also impacted participant experiences. I4 summarized social support as paramount:

> *You need both individual buy-in and you also need organizational buy-in from the university. Otherwise you can actually have the best and brightest student in the class who happens to be blind and if they can't get the requisite support, then they've got a huge, huge, huge problem. - i4*

As such, we briefly overview the impact of faculty and other students on participants' educational experience.

Faculty were often responsible for accommodating students directly, most commonly by sending course materials even when they did not typically distribute them to motivate class attendance. Participants reported incidents when faculty would refuse to send these materials and i4 had to enroll the assistance of university compliance personnel when advocating for himself did not work. When faculty were unwilling to work with the students, it made it difficult for students. Five participants (i1, i2, i3, i4, i5) mentioned having push back from the faculty or dropping courses to have a more accommodating faculty member. In particular, i2 had faculty that encouraged him to change his major and he nearly dropped his degree because he was frustrated with having to fight to gain access to materials. However, participants also praised supportive faculty; i1 remembered one faculty member who sent slides before each lecture, complete with text descriptions of diagrams.

Accessibility barriers also manifested during group projects. Several projects required students submit diagrams of how the software will work. I6 recounted his team leaving him out since he couldn't discern diagram drawings based on conversation alone

> *But even so, it wasn't ideal. I mean they were understanding, but, you know, there were lots of sessions where they'd all be talking and I wouldn't be able to say anything at all. Not because I didn't want to, but just because I couldn't really follow what they were talking about. - i6*

Participants had difficulty following real-time creation of inaccessible materials to be submitted for their group projects. This had further consequences than just their participation in the work the group was doing. I6 continued, *"But it's also a social thing, you know. Cause if you're the one person in the group that doesn't talk, you know, guess who the one person who doesn't get* *invited for drinks afterwards is."* Being unable to participate in parts of the group projects was isolating for some participants. As other students can be help one another study and navigate computer science altogether, participants who did not contribute equally during group projects experienced isolation.

5. Discussion

We uncovered numerous barriers that characterize computer science education for blind students. Our participants recalled exploring numerous alternative technologies, textbooks, and assignment formats which added extra work. Further, accessibility barriers fostered disbelief that new technologies would be accessible, decreasing motivation to engage with important tools and concepts they would later need in their careers. Finally, social factors were hugely important but under addressed. We'll now highlight three areas for collaboration among educators and researchers toward removing these barriers.

5.1 Enforce Accessible Course Materials

First, we believe that enforcing the use of accessible course materials will alleviate pressure for blind students to explore accessible alternatives. Throughout our interviews, it was clear accommodations often remediated tools that could have been designed more accessibly. To enforce use of high quality accessible materials and tools, it will require partnerships with faculty, curriculum developers, education technology researchers and companies, university IT, and offices for students with disabilities. Faculty will need to be involved to provide subject matter expertise, university personal who prepare and implement accommodations for students with disabilities will provide more expertise on specific issues faced by students with disabilities and the creation of the alternative materials. Additionally, if curriculum developers and educational technology research consider the needs of students with disabilities from the beginning, then it can decrease the need for accommodations.

Further research is needed to improve the process of creating accessible materials. Currently, the central challenges include time, expertise, and specialized equipment needed to provide accessible alternatives. This not only burdens students but requires faculty to have their materials ready far in advance. Alternative materials production then rests on the few experts, removing agency from blind students and faculty until accessible materials are sent back in return. There has been some work starting in this area to help improve the process of creating tactile formats of the images in a textbook using the Tactile Graphics Assistant [7]. However, this is just one piece and we need to continue improving alternative materials production to be expedient and producible by non-experts. This work can uplift everyone involved in providing accommodations by allowing students and faculty to be able to play a more active role.

5.2 Support Faculty Learning

Second, we found that faculty need more support. Blind participants were largely the only information source for faculty. Yet blind students are not experts in accessible education, and when left to retool inaccessible curriculum, faculty can be further burdened or even suggest damaging alternatives like exempting assignments. AccessComputing [4] aims to equip instructors with Universal Design for Learning techniques such that students with disabilities can engage with few accommodations. Faculty and disability service professionals can also participate in communities of practice to create networks that can help identify steps to make classrooms and programs more accessible and inclusive and recruit more students with disabilities [22]. Researchers must consider how to scale these trainings and approach departments and universities for greater buy-in so faculty participation is expected and rewarded.

5.3 Foster Social Integration of Blind Students

Finally, blind students should be intentionally socially integrated. Accommodations to functionally be able to complete assignments are viewed as the means for disabled students to access their education. Yet we found these modifications were insufficient and could even increase isolation when, for example, an alternative assignment meant a participant was the only person working with a partner or building a different system altogether. First, blind students should have access to the spaces, machinery, and events hosted by departments. Recruitment events to improve diversity should explicitly include disability as an underrepresented group. Finally, adapted coursework should minimize social isolation and assignments deemed too visual for blind students should be replaced for the entire class with more inclusive alternatives.

Further research on creating more accessible technologies, such as collaborative accessible graph creation, could increase engagement from blind students during group work and other assignments. As technologies become more accessible, there will be less burden on students to find good accommodations.

6. Conclusion

This study shows that blind students encounter numerous barriers to obtain a computer science degree, and current research and efforts are only scratching the surface. Existing programs must be amplified and supplemented with more direct support from educational technology researchers and universities to remove barriers. In this paper, we revealed specific obstacles blind students faced and posited three overlapping areas for collaboration among educators and researchers to foster more fair and welcoming programs. By implementing accessible course materials and tools, supporting faculty accessibility education, and prioritizing integrating blind students, we hope to provide some steps forward toward a more diverse computing field.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. (DGE-1256082).

REFERENCES

[1] Catherine M. Baker, Lauren R. Milne, and Richard E. Ladner. 2015. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 3043–3052. https://doi.org/10.1145/2702123.2702589

[2] Suzanne P. Balik, Sean P. Mealin, Matthias F. Stallmann, Robert D. Rodman, Michelle L. Glatz, and Veronica J. Sigler. 2014. Including Blind People in Computing Through Access to Graphs. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '14*, 91–98. https://doi.org/10.1145/2661334.2661364

[3] Jeffrey P. Bigham, Maxwell B. Aller, Jeremy T. Brudvik, Jessica O. Leung, Lindsay A. Yazzolino, and Richard E. Ladner. 2008. Inspiring Blind High School Students to Pursue Computer Science with Instant Messaging Chatbots. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education - SIGCSE '08*, 449. https://doi.org/10.1145/1352322.1352287

[4] Sheryl Burgstahler. 2008. *Equal access: {Universal} design of instruction.* Retrieved November 13, 2018 from https://www.washington.edu/accesscomputing/sites/default/files/doit-sync/files/Equal-Access-Universal-Design-of-Instruction.pdf

[5] Sheryl Burgstahler and Richard Ladner. 2006. AccessComputing: From Research to Practice | The Alliance for Access to Computing Careers. Retrieved May 3, 2016 from http://www.washington.edu/accesscomputing/resources/accesscomputing-research-practice

[6] Parham Doustdar. 2016. The Tools of a Blind Programmer – Parham Doustdar's Blog. Retrieved April 21, 2016 from https://www.parhamdoustdar.com/2016/04/03/tools-of-blind-programmer/

[7] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. 2007. Automated Tactile Graphics Translation: In the Field. *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility - Assets '07*: 75. https://doi.org/10.1145/1296843.1296858

[8] Maggie Johnson. 2015. The Computer Science Pipeline and Diversity: Part 1 - How did we get here? Retrieved May 7, 2016 from http://googleresearch.blogspot.com/2015/07/the-computer-science-pipeline-and.html

[9] Shaun K. Kane and Jeffrey P. Bigham. 2014. Tracking @stemxcomet: Teaching Programming to Blind Students via 3D Printing, Crisis Management, and Twitter. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14*, 247–252. https://doi.org/10.1145/2538862.2538975

[10] Stephanie Ludi. 2013. Robotics Programming Tools for Blind Students. In *28th Annual International Technology and Persons with Disabilities Conference Scientific/Research Proceedings.* Retrieved May 26, 2016 from http://scholarworks.calstate.edu/handle/10211.3/121968

[11] Sean Mealin and E Murphy-Hill. 2012. An Exploratory Study of Blind Software Developers. In *Proceedings of 2012 IEEE Symposium on Visual Languages and Human-Centric Computing - VL/HCC '12.* Retrieved February 27, 2014 from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6344485

[12] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y. Vidya, Manohar Swaminathan, and Gopal Srinivasa. 2018. CodeTalk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 1–11. https://doi.org/10.1145/3173574.3174192

[13] T. V. Raman. 1996. Emacspeak---Direct Speech Access. In *Proceedings of the Second Annual ACM Conference on Assistive Technologies - Assets '96*, 32–36. https://doi.org/10.1145/228347.228354

[14] Jaime Sánchez and Fernando Aguayo. 2005. Blind Learners Programming Through Audio. *Extended Abstracts on Human Factors in Computing Systems - CHI '05*: 1769–1772. Retrieved February 27, 2014 from http://dl.acm.org/citation.cfm?id=1057018

[15] Ann C. Smith, Joan M. Francioni, Mohd Anwar, Justin S. Cook, Asif Hossain, and M. Fayezur Rahman. 2003. Nonvisual Tool for Navigating Hierarchical Structures. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '04*, 133. https://doi.org/10.1145/1029014.1028654

[16] Andreas Stefik, Christopher Hundhausen, and Robert Patterson. 2011. An empirical investigation into the design of auditory cues to enhance computer program comprehension. *International Journal of Human-Computer Studies* 69, 12: 820–838. https://doi.org/10.1016/j.ijhcs.2011.07.002

[17] Andreas M. Stefik, Christopher Hundhausen, and Derrick Smith. 2011. On the Design of an Educational Infrastructure for the Blind and Visually Impaired in Computer Science. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*, 571. https://doi.org/10.1145/1953163.1953323

[18] Paul Vickers and James L Alty. 2002. When bugs sing. *Interacting with Computers* 14, 6: 793–819. https://doi.org/10.1016/S0953-5438(02)00026-7

[19] About | The Alliance for Access to Computing Careers. Retrieved April 21, 2016 from http://www.washington.edu/accesscomputing/about

[20] EdSharp. Retrieved November 8, 2017 from https://github.com/EmpowermentZone/EdSharp

[21] Learning Ally - Support for Dyslexia and Learning Disabilities. Retrieved May 7, 2016 from https://www.learningally.org/

[22] Communities of Practice | AccessComputing. Retrieved December 1, 2018 from https://www.washington.edu/accesscomputing/get-involved/educators-employers/communities-practice