University of Glasgow | School of Computing Science

Investigating the Relationship Between Spatial Skills and Computer Science

Jack Parkinson (2082657)

July 2, 2018

ABSTRACT

Spatial skills, also known as spatial ability, spatial reasoning or spatial cognition, have been connected with STEM and computer science success for over sixty years. While in some fields, notably engineering, this relationship has been well explored and exploited, there is limited research investigating the relationship between spatial skills and computer science. The research that does exist indicates a correlation between spatial skills and computer science success, however, a cognitive model suggesting why this relationship exists has not been proposed. Unlike Sorby's work in engineering, we do not believe there is sufficient evidence to launch into full-scale spatial skills training for our students, but yet are tantalised by the possibility that this may be effective in improving CS skills, enough to warrant a small scale pilot study. The aim of this paper therefore is to provide a more solid foundation concerning the connection between computer science and spatial skills for future studies. It surveys the literature on spatial skills and investigates the various underlying cognitive skills involved. It then poses a theoretical model for the relationship between computer science ability and spatial skills, exploring ways in which the cognitive processes involved in each overlap, and hence may influence one another. This is strengthened by an experiment reported here involving 72 participants, determining how spatial ability relates to computer science attainment, the results of which suggest that spatial skills typically increase as the level of academic achievement in computer science increases. Further, it presents the findings of a pilot study, which replicates a small scale spatial skills training course to investigate the effects of training on students and identify the practical considerations which need to be made in preparations for running such a course on a larger scale. The background research, model and pilot study detailed here support future experiments to determine whether spatial skills training can lead to improved computer science skills.

1. INTRODUCTION

Skills in STEM subjects appear to be related to spatial skills (SS): STEM practitioners are reported to have high SS, relative to others [44]; training in SS can improve abilities in STEM subjects, particularly engineering [33]. There is tantalising evidence of such a relationship in computer science (CS), which, due to the cheap and easily accessible nature of SS training, could lead to higher achievement and lower dropout rates, as with engineering.

Unfortunately, current studies in this area are limited and inconclusive - correlation has been identified [18], but only one study [43] firmly shows that SS training appears to help in computing. Based on this inviting start, further study is warranted.

SS are not easy to define strictly [39], and as such studies contain unclear and contradictory descriptions which are likely to hamper research efforts. Perhaps as a result of this, no studies hypothesise why the STEM and SS relationship exists to any great extent, and so current work may not be optimally focused. Furthermore, most studies in the field tend to concentrate on a single cohort, typically entry level students, without examining effects across experience levels.

Based on these gaps, this paper presents three main additions to the research in the field. First, I summarise what is known about SS, defining core elements of SS and how they can be measured. Second, I propose a model for the relationship between SS and CS, drawing on key cognitive processes which appear to be shared by both fields. Third, I describe an experiment to examine the relationship between SS and CS attainment across a range of CS practitioners, from entry level students to professors.

This paper also details a pilot study in which a course was delivered to train the spatial skills of entry level computing science students. The purpose of this course was to investigate the effects of training the spatial skills of first year students on their computing aptitude. In addition to examining this relationship statistically, a further purpose of the course was to identify elements leading to success and the practical aspects of running such a course.

These deliverables are valuable contributions to our understanding of this area, particularly for laying a stronger foundation for future experiments to examine whether or not, and how best, computing ability can be improved with SS training.

2. RELATED WORK IN SS AND STEM

Spatial skills have been connected with STEM for almost seventy years, since Super and Bachrach examined the skills of mathematicians, engineers and natural scientists, and found SS to be a factor in all these fields [38]. In a broad study covering the work of dozens of researchers, Super and Bachrach attempted to classify the skills and traits of professionals in science and engineering, reviewing studies on such factors as mathematical ability, verbal ability and several other "special" abilities, including SS. They found that not only is SS prominent in these fields, but that in cases where the relationship was tested STEM practitioners outperformed non-STEM people in SS tests, even those recognised as being "gifted" in other fields.

Wai *et al.* undertook an investigation on SS pertaining to

Project TALENT data [44, 45]. Project TALENT consisted of a series of tests given to over 400,000 high school students in the US in 1960 and subsequent follow up questionnaires up to the 1970s. Of the students who went on to achieve a PhD in a STEM field, most scored highly in the Project TALENT spatial skills tests taken eleven years previously (with 45% being in the top 4% of SS scores). Again, the relationship is not causational; SS are shown only to be correlated to progression in STEM subjects.

The STEM area with most research relating to SS is engineering. Sorby has investigated this relationship for over 20 years, showing that engineering students who receive SS training do better in their engineering courses and have lower dropout rates [33]. In addition to developing a SS training course, consisting of a workbook and software [35], Sorby has shown positive effects of training SS initially on self-selecting groups of low SS scorers in engineering, and then a similar effect in compulsory courses provided by Michigan Tech [32]. The effect of these studies are significant and well replicated: one can reliably train SS to see an improvement in engineering success.

SS also have relationships with success in other STEM fields. In physics, Kozhevnikov *et al.* discovered that psychology undergraduate students with better spatial visualisation skills performed better in, and were able to more clearly explain, kinematic physics problems [19]. Pallrand and Seeber conducted a separate examination in physics, and noted that not only did students undertaking a physics course show higher gains in spatial skills compared with liberal arts students on pre/post tests, an experimental group undertaking additional SS training outperformed the placebo and control groups [25]. This study is similar to studies undertaken by Sorby, showing the effectiveness of a training course which can be taken alongside standard teaching [33]. Crucially, it also shows that SS can be developed during study of a STEM subject without dedicated SS training, a point to be returned to later in section 4.

Carter *et al.* showed that those with higher SS outperformed those with lower SS in a general chemistry course [3], and have also been found to have an effect in organic chemistry when specifically required to manipulate and understand representations of molecular molecules [26].

In mathematics, Tartre identified that spatial orientation ability is applied in certain mathematical problems, and suggested that the ability was specifically related to particular mathematical skills, such as determining area of irregular shapes and groupings of associated objects [39]. However, Tartre's chosen test for spatial orientation is more typically used as a test of closure speed [9], and one of the selected mathematical problems is very similar to an existing test of spatial relations (shown in figure 4). Another study indicating a connection between spatial visualisation and mathematics was conducted by Fennema and Sherman, who showed that spatial skills are one of the factors contributing to the gender gap found in mathematics [10].

In addition to these studies, Veurink and Sorby [43] have shown that the training course developed by Sorby and Baartmans [34] (and consequentially developed into a training workbook [35]) can be used to potentially improve the results of engineering students undertaking non-engineering modules. Several cohorts of engineering students taking additional modules (in areas such as calculus, physics and chemistry) had their SS measured at the start of the course. Those who failed a SS test were offered a chance to increase their SS by taking additional training, and ultimately these students did better in their respective elective modules than their peers who also failed the test and opted not to take the training [43].

Note that these studies are but a few selected as highlights to illustrate the breadth of the relationship between spatial skills and STEM fields and are not exhaustive. Other studies in chemistry [5, 20], mathematics [12, 13, 16, 41], physics [15, 21], engineering [46] and STEM in general [23] have found correlations with spatial skills and success, but have not been fully explored in this paper

In Veurink and Sorby's paper, another module in which students excelled after SS training was a computing module, specifically introductory programming. Students who initially failed the SS test and opted to take training showed significantly higher GPAs in their computing course than those who failed or marginally passed the test, but did not take additional training. This result is based on 6 cohorts, totaling 74 participants, of self selecting students between 1996 and 2002. This implies a causal relationship from SS to programming, but note that self efficacy cannot be ruled out as a factor in these findings, since students self-selected to take the additional training, and it is possible that the students who have a more proactive attitude were both likely to take the course when offered and do better in their elective modules anyway. Additionally, there was no prior measure of computing ability, which could be a confound in the study.

Though not making reference to the study by Veurink and Sorby, Cooper *et al.* attempted to show a similar result [6]. It is strange that Cooper's study, of which Sorby is a co-author, does not reference this earlier, apparently highly-related, work. Cooper took a selection of summer school students intending to begin a university course in computing, and over a period of two weeks, trained their SS in an experimental group and compared their gains in a standardised computing test. The authors acknowledge some issues with the study, e.g. the questions used to test computing ability may not have been the most effective for the group of students they had. The increase in gains by the experimental group failed to reach significance, except when the six questions from the test with the highest item discrimination were selected. Ultimately, the authors clearly state that they are not claiming causation, but a correlation which requires further research.

A similar correlation was displayed a few years earlier by Jones and Burnett [18]. They took a cohort of Masters conversion students who had not previously studied computing, tested their SS and examined their end of year results. They did not see any correlation in the Introduction to Human Factors or the IT Management courses taken by these students, but did see a correlation between the Introduction to Programming course and the Object Oriented Systems course, both of which required a significant amount of programming. This suggests that it is possible that the connection with SS lies not strictly with computing generally, but specifically with programming.

A large, multi-institutional study undertaken by the University of Kent identified a similar correlation [11]. It was found that spatial visualisation showed a correlation with success in CS, as did a progression of map drawings - that is, those who were capable of successfully drawing more complex maps (with landmark maps being considered relatively simple compared to survey maps, for example) tended to do better in CS. While map drawing is not strictly considered to be a dedicated factor in spatial skills, there are certainly related cognitive processes involved, such as the ability to visualise and consider space from different perspectives - see section 3 for a more in-depth description of the factors contributing to spatial skills.

Based on this existing research, we highlight two points. First, evidence of a causal relationship between SS and CS is limited, though there is certainly something of interest in the area. Second, no researchers have attempted to examine this relationship in an attempt to explain *why* it might exist to any significant degree. In an effort to remedy this, this paper shall attempt to lay theoretical groundwork for the existence of such a model. It is my belief that a stronger understanding of SS and how they relate to CS will help researchers to pinpoint the effect of SS training and what gains it may provide in a computing context.

3. UNDERSTANDING SPATIAL SKILLS

Spatial skills is a broad term lacking a concise definition, and as such making clear, distinct arguments about SS can prove difficult. Tartre effectively surmises problems faced in discussing and communicating spatial ability and their impact:

> Attempting to understand and discuss something like spatial orientation skill, which is by definition intuitive and nonverbal, is like trying to grab smoke: The very act of reaching out to take hold of it disperses it. It could be argued that any attempt to verbalize the processes involved in spatial thinking ceases to be spatial thinking [39].

In an effort to reduce ambiguity and overcome issues pertaining to the rift between written descriptions of SS and their practical applications, various tests of specific SS factors are presented when they are introduced into discussion. Most of these tests have been extracted from Ekstrom *et al.*'s manual for factor-referenced cognitive tests [9].

Over years of discussion and exploration of this difficult field, Carroll collates a wealth of research into a cohesive model consisting of the following factors [2]:

- Spatial Visualisation
- Spatial Relations
- Closure Speed
- Closure Flexibility
- Perceptual Speed
- Visual Imagery (though Carroll identifies this factor as a theoretical factor, without coming to a clear conclusion on its definition)

Spatial visualisation is the factor that has been most examined in relation to STEM, including CS. McGee identified spatial visualisation prior to Carroll as one's proficiency in being able "to mentally rotate, twist, or invert pictorially presented visual stimuli" [22]. Tartre presents a substructure of two distinct factors contributing to spatial visualisation: mental rotation and mental transformation [40]. Mental transformation involves the manipulation and modification of objects, required for such practical applications as visualising cross sections or intra-part movements. This can be seen in practice in the Mental Cutting Test (MCT) [4] in figure 1. Mental rotation is the ability to perform rotations on mental constructs. Practically this typically translates to the ability to see a physical representation of a structure (a block on a table or an image on a piece of paper) and mentally imagine what this object or shape would look like rotated in a different orientation. Ho and Eastman discovered that 2D and 3D rotations are closely related, supporting Carroll, but also that one capable of performing 2D rotation may not be capable of performing 3D rotation [17]. An example of a test of 2D rotations is displayed in figure 2 [44], and an example of a 3D rotation test can be found in figure 3 [47].

Figure 1: Test of mental transformation, the Mental Cutting Test - *identify the cross section after the following transformation has occurred*

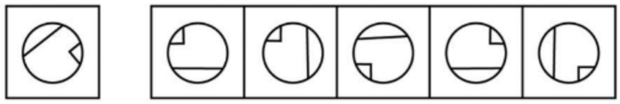

Figure 2: Test of 2D mental rotation - *which of the following corresponds to the original shape*

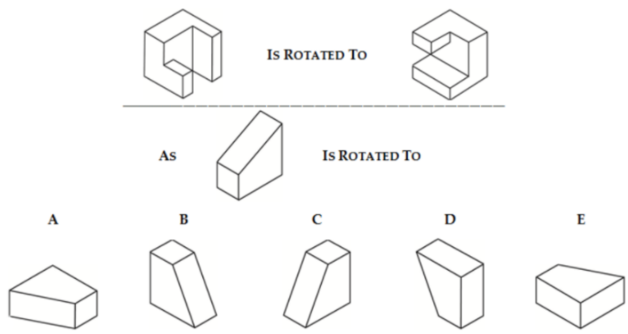

Figure 3: Test of 3D mental rotation, the revised Purdue Spatial Visualisation Test of Rotations (PSVT:R)

Mental rotation is related to another core factor of SS: spatial relations, which is the ability to understand the arrangement and orientation of objects or patterns within their environment. While this initially appears very similar to mental rotation, spatial relation applications do not strictly require rotation to take place, merely a decent understanding of object orientation. In practice, a test used to measure spatial relations is the Cube Comparison Test, displayed in figure 4 - as can be seen, to find the correct answer, objects do not need to be rotated (which in fact, would be difficult to do with the lack of information of the object); the examinee just needs to be able to relate each face of the cube to its neighbours.

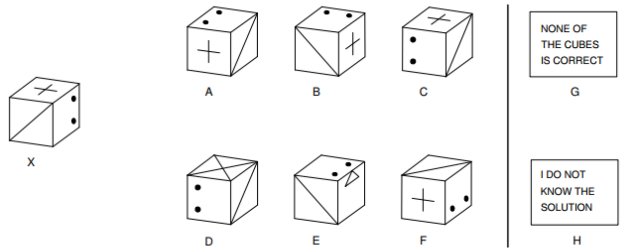

Figure 4: Test of spatial relations, the Cube Comparison Test

Closure speed, closure flexibility and perceptual speed can be defined as follows:

- **Closure Speed:** speed in identifying an *unknown* pattern from an *obscured* environment
- **Closure Flexibility:** speed in identifying a *known* pattern from an *obscured* environment
- **Perceptual Speed:** speed in identifying a *known* pattern from an *unobscured* environment

The easiest way to perceive the application of these these skills is by examining the tests associated with them. Closure speed is measured by the Gestalt Completion Test [37] (figure 5), which requires the test subject to pick out some representation of an object or image from a highly distorted image. Closure flexibility can be tested by the Hidden Figures Test [9] (figure 6), in which the test subject is provided with a selection of figures (which are *known*) and a complex pattern, and are required to identify which of the given figures is obscured within the pattern. Perceptual speed is tested by the Identical Pictures Test [9] (figure 7), in which the test subject is presented with a figure and a lineup consisting mostly of figures *similar* to the given figure, with one figure being identical. The test subject is required to identify the figure from the lineup matching the one provided.

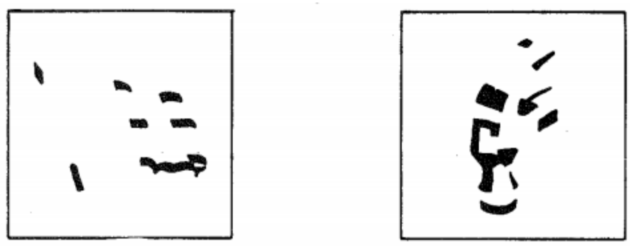

Figure 5: Test of closure speed, the Gestalt Completion Test

Carroll also identifies a final first order factor of SS as visual imagery. Visual imagery is a somewhat vague factor in the discussion of SS, and lacks the definition and clarity of other first order factors of SS. Burton and Fogarty attempted to measure this factor, and ultimately decided that the best model they constructed was one which included three second order factors contributing to visual imagery [1]. These are:

- **Quality:** "the ability to generate, maintain, and transform a clear visual image"

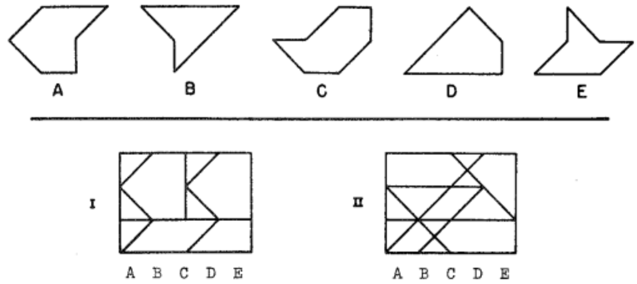

Figure 6: Test of closure flexibility, the Hidden Figures Test

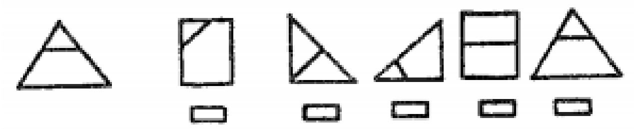

Figure 7: Test of perceptual speed, the Identical Pictures Test

- **Self-report:** "ability to generate, control, and/or rotate a visual image"
- **Speed:** "latency measures derived from the experimental tasks" - that is, the tasks which were used to determine the existence of the above two factors

These factors fit into spatial skills beneath the term visual imagery, contributing to the theoretical factor which Carroll identified.

4. MODELING SPATIAL SKILLS AND CS

With an understanding of spatial skills and the factors contributing to them, we can now attempt to show their connection to CS. It is noted that existing studies relating SS and CS have focused on programming, and I recognise that the underlying skills in programming, such as the development and manipulation of models and the ability to represent these textually and graphically, are core skills across much of CS. Hence I too will focus on aspects of programming.

A fundamental ability in programming is program comprehension. Much research has gone into examining methods and cognitive frameworks involved in program comprehension [29]. One such model is the model of the schema, presented by Détienne and Soloway [7]. A schema is a kind of data structure stored in memory which represents some construct: it consists of a plan, which is some generic process or operation as the user understands it; and cues, which are points of reference used to match up a plan with an associated function. In practice, an application of a schema may consist of identifying key variable declarations or structures in code (such as MAX or COUNT, or the beginning of a loop), matching them with associated schema and applying a plan which is applicable.

The schema model is of significance because operations involved in building a schema can be mapped to SS operations. The identification of cues requires that patterns be extracted from obscured environments, not unlike the process required in the application of closure flexibility. These cues are pointers to a model or structure which must be constructed mentally in order to formulate a process. This is similar to several exercises in Sorby's workbook involving the composition of isometric 3D objects from a selection of 2D orthographic views, taking note of specific, useful data points and constructing a more complex structure combining this data (an example of such an exercise can be seen in figure 10).

Another code comprehension framework is the Block Model proposed by Schulte [30]. The Block Model involves a process of examining code at four levels, to identify (1) atoms (single words or simple statements in the code), an understanding of which is used to construct (2) blocks ("regions of interest that syntactically or semantically build a unit"), (3) relations (connections involving blocks and atoms such as a *find maximum* code section) and (4) the macro structure (the overall operation of the program). The method of building up from atoms to blocks and relations is similar to Détienne and Soloway's process of schema construction, and likely requires the same cognitive processes, again relating to the application of spatial skills.

A further element of the Block Model of particular interest is the choice of the name for the third stage in the structure of the program being examined: relations. This stage consists of taking the blocks and atoms recognised previously and understanding how they relate to one another, without strictly having to perform operations on the blocks themselves (since an understanding of these has already been developed). If the objects in question were cubes or shapes, this definition would be very close to that of spatial relations. It is not unlikely that cognitive operations required for both applications of understanding relations are related, be they between pictorial representations of objects or expressions in code.

Another important aspect of program comprehension is the idea of the notional machine, first identified by du Boulay as a combination of knowledge - of the programming language, environment and data - and a mental model [8]. Sorva describes the function of a notional machine as "an idealized abstraction of computer hardware and other aspects of the runtime environment of programs." [36] Sorva closely connects the ability to form notional machines, and therefore appropriately and effectively comprehend programs, with the ability to construct an abstract mental model. Experts develop more robust, adaptable mental models than novices, whose mental models tend to be "fragile". Sorva also discusses the "runnable" nature of a mental model, based on the work of Norman [24], which involves the user being able to "envision with the mind's eye how a system works," and directly associates this with working memory and visualisation.

When reviewing spatial skills factors, there are only two which match up with this process of forming a mental model: spatial visualisation (as Sorva briefly suggests) and spatial relations. Closure speed, closure flexibility and perceptual speed are all related to identifying patterns from environments, and visual imagery relates to capturing and recalling images, leaving the two aforementioned factors. An element of spatial relations would be required to construct a mental model, as the user requires an understanding of how various components are linked together (of how they *relate*), but spatial visualisation provides more robust abilities for these tasks. A robust mental model must be subject to development and restructuring as required - the ability to perform these actions mentally is closest to mental transformation (the ability to manipulate or modify a structure mentally) which is part of the spatial visualisation factor. An element of spatial relations may also be included, but typically spatial relations consist of a simple inter-object understanding (see figure 4 for an example) compared with mental rotation, which requires a deeper understanding of the constructs involved (see figure 3). This indicates that when trying to understand more complex constructs in a mental model and what they would look like in a different orientation or situation, spatial relations are likely to work to an extent, but the more complex operations are more likely to require mental rotation (another subset of spatial visualisation).

A difficulty here arises with the definitions of spatial skills factors as given in the literature. From the CS side, considering mental models, we are constructing a mental representation of some operation or process. However, this does not directly relate to a specific factor of spatial skills, forming a neat, clear connection between the two. I identify the closest match as spatial visualisation, where typically the same ability to construct a mental structure is required before then performing some operation on it, such as a rotation or transformation. As such, I theorise that spatial visualisation is very likely to contribute to program comprehension in this regard.

In addition to program comprehension, another core aspect of computing is the procedure of program generation. While generation must be closely related to comprehension, as any generation plan must also involve a process of debugging and review [29], there are elements of program generation not included in comprehension.

Rist observed a method of program generation which he named "focal expansion" [27]. The process of focal expansion involves reviewing a problem and identifying a core function or plan on which to base the implementation. The process which follows involves taking the core plan and building outward, adding and expanding as necessary to facilitate the generation of a program that fully satisfies the problem. Rist links this process to working memory, and associates the ability to track the program generation mentally, from the focal point out to the full solution, with working memory capacity [28]. While this does appear to be the case, it is also possible that visualisation factors into the programmer's ability to track the expansion: to quote Sorva again, "to envision in the mind's eye." Also pertinent to program generation is problem comprehension, the process of identifying a problem from some specification - this process is similar to the schema process of identifying a plan in practice, except that rather than looking for cues in a program they must be extracted from a problem description.

Cues are a recurring concept in both program comprehension and problem comprehension which has briefly been touched on. This involves the process of identifying potential patterns from a broader environment consisting of more details than the user is currently interested in. There are also factors of SS which, in practical use, are used in performing a very similar task to these operations: closure speed, closure flexibility and perceptual speed. Recall that these processes involve the extraction of patterns (known or unknown) from environments (obscured or unobscured).

The simplest factor is perceptual speed, which is a process of identifying a known pattern from an unobscured environment - though rare, this may have an application in cue identification. In code comprehension a case may arise where the user knows the construct they are looking for and the code is laid out in such a way that there is minimal interaction and obscurity between lines (an example of this may be looking for a known variable name in a list of declarations, such as at the top of a file). More likely, the user will be searching for an pattern to match against a record of terms which they feel may have significance based on prior knowledge (such as the start of a loop or declaration of some telling variable). Pictorially, this would be very similar to the test in figure 6, so it is possible that the cognitive process involved in closure flexibility is relevant to this style of program comprehension. And finally we have a case for closure speed, in which an unknown pattern must be derived from an obscured environment. This is akin to an application of the schema model, in which the user searches the code space to identify cues which are not previously known to match them to known schema - at least this would be the case for experts; novices are likely to take a different approach which will involve less searching and pattern matching and more mental model construction.

In this section so far, we have analysed significant aspects of CS and connected them to SS, forming the elements of a model, a diagrammatic representation of which is presented in figure 8.

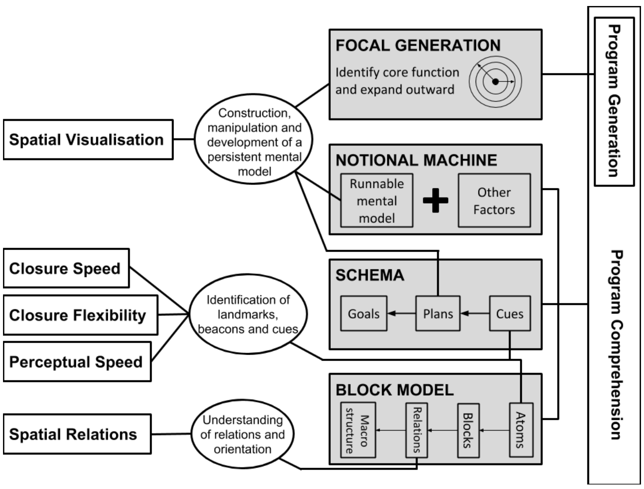

Figure 8: Diagram of the relationships observed between spatial skills and computing science

Bearing this model in mind, I shall now discuss the implications for the development of SS and what this could mean for CS. Sorby notes that the most effective method of training spatial skills is by hand sketching diagrams and drawings [31]. This can be seen in action in her workbook, which poses dozens of short form drawing questions to be completed over a relatively short period of time. It is our theory that the reason why spatial skills are connected with computing is because *the same* cognitive functions are involved in computing and also in other more obvious applications of spatial ability - as a result, SS training could affect one's computing ability positively, just as Pallrand observed in physics [25]. However, with this model one can also deduce that training in CS develops SS. In which case, why would SS training be of any benefit when the same could be achieved with a standard computing course?

It is my theory that SS training, such as Sorby's workbook, is far more focused and directed than typical programming courses. Whereas in an entry level programming laboratory students may be expected to write a handful of short programs to achieve given goals over the space of a couple of hours, Sorby's exercises can consist of up to forty sketches to be completed in about half the time. Furthermore there far fewer barriers to advancement: any given drawing could be attempted regardless of the student's experience (a complete novice who has never done any spatial skills training could pick up Sorby's book and attempt the questions), compared with a programming student who must first learn code snippets required for tasks before they can be reused in later tasks. It is possible that the same skills are being developed, but at a slower rate for programming students who must learn not only how to think computationally, but how to write code properly. It is also possible that the students who fall behind in programming are the ones whose SS are not as developed as their peers, and the barriers to their progression are rooted in their inability to construct robust and adaptable mental models (key to programming comprehension and generation).

With this in mind, it is theorised that spatial skills themselves do not directly contribute to computing science or other STEM domains, but rather than the cognitive functions involved in SS are also involved in STEM domains: such functions as the ability to form, manipulate and develop mental models, identify key points in an environment and understand relations between structures. Based on this theory, we present a simple model for this relationship in figure 9.

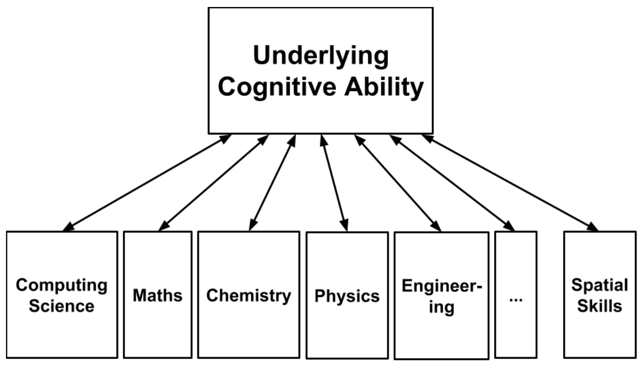

Figure 9: Relationship between cognitive functions behind spatial skills and STEM domains

Notice that the relationships between domains and the underlying cognitive ability are bi-directional. As stated, I believe that these cognitive skills can be developed by pursuing a STEM domain or by training SS, however due to the direct and precise nature of spatial skills training, this route is likely to produce results more effectively. Moreover, training this ability in one area is likely to have an effect on other areas which make use of the same skills: by training spatial skills, we may see one's ability to write or understand programs improve.

5. SPATIAL SKILLS AND CS ATTAINMENT LEVEL

In previous experiments and studies of SS and CS, it has been typical to examine students either in their first year of study in computing (as an undergraduate degree, a Master's degree or an elective module) or about to start the former. Jones and Burnett's study suggests that those who are better at programming are likely to have better SS, but this is only across a single year.

To better understand the relationship between SS and CS beyond a single year, the SS of students and staff at different levels of attainment in the department of Computing Science at the University of Glasgow were measured. The study involved two research questions:

- **RQ1:** Do spatial skills vary with academic attainment in CS?
- **RQ2:** Do spatial skills vary with areas of specialisation in CS?

For **RQ1**, given Jones and Burnett's results, it was expected that the higher the attainment and so the "better" the CS skills, the higher the SS will be.

RQ2 draws on Jones and Burnett's finding that SS are not significantly connected to non-programming courses. Each test participant was asked to record their specialised or most favoured area of computing. With this data, it was expected that those involved in heavily programming oriented areas of CS - such as software engineering or systems development - would overall have higher SS than those who were focused on more human based courses, such as HCI or human-centered security.

Furthermore, studies have indicated that gender can affect the SS of participants [10, 42]. In order to account for this, the gender of each participant was also recorded.

5.1 Method

5.1.1 Participants

Five cohorts were selected to draw participants from:

- First year students, specifically taking CS1CT, the introductory CS route designed for those without programming experience
- Honours students from third and fourth year, who predominantly take the same courses at the same level
- MSci students, who were taking an undergraduate Master's course immediately after their fourth year
- PhD students
- Staff from the computing science department

Participants were selected at random and then approached to take the spatial skills test, with the exception of the first years, who all took the SS test during a lecture and 30 were selected at random for the study.

Participants were also required to indicate their specialised or preferred area of research. To reduce the granularity of the data, the participants were grouped according to which of the institution's computing science research sections they fell under. The department has four main research sections which:

- **GIST:** focusing on human-computer interaction and human factors
- **IDA:** focusing on machine learning, information retrieval and data science
- **GLASS:** focusing on systems engineering and networks
- **FATA:** focusing on algorithms, computational thinking, formal analysis and mathematical modeling

Note that the first year students were not required to indicate a preferred area of computing. Additionally, some participants opted not to provide this information. Table 1 details the breakdown of the participants.

Level	Male	Female	GLASS	GIST	IDA	FATA	Total
Level 1	19	11					30
H Level	9	4	4	1	3	1	13
MSci	7	2	1	0	4	3	9
PhD	5	5	3	3	1	3	10
Staff	8	2	2	1	1	5	10
Total	43	29	10	5	9	12	72

Table 1: Table displaying the characteristic breakdown of participants

5.1.2 Test delivery

The test used was the Revised PSVT:R [47]. For all intents and purposes the Revised PSVT:R consists of the same questions as the original PSVT:R by Guay [14], but has been updated to have some graphical errors fixed and the questions arranged in order of difficulty. The test was conducted with a 20 minute time limit.

The test was provided in two separate formats: online and on paper. Online, the test was accessible through the institution's Moodle platform via a quiz with a 20 minute time limit, after which no further answers could be given. On paper, an overseer was present to ensure that the test was completed within the time limit and that the participants were not looking up answers or conferring. While we cannot absolutely confirm that those who completed the test online did not confer, the timer was not pausable and once the test was begun could not be reset, so in order to cheat participants realistically would have had to have begun the test with the intention of doing so. In addition to the fact that the answers to the PSVT:R are not readily available online, we do not expect that any of the participants would have attempted to invalidate the research deliberately.

Once the tests were completed, the scores were collated along with level of attainment and demographic data. By this stage of the research, no names or other sensitive data was attached to any of the results.

5.2 Analysis of Results

Once the data had been collected, the mean and standard deviation for each group was calculated and are displayed in table 2.

After breaking down participants into groups based on attainment level and gender, the SS means of these groups are displayed in table 3.

To confirm the validity and significance of the experiment, a two way analysis of variance (2-way ANOVA) was conducted. The results of this statistical method are displayed

	Level 1	H Level	MSci	PhD	Staff
Mean	18.97	22.92	24.67	22.00	25.50
SD	6.21	4.59	4.00	6.13	3.60
n	30	13	9	10	10

Table 2: Table displaying the mean, standard deviation and number of participants for each cohort

	Level 1	H Level	MSci	PhD	Staff	Total
Male	20.42	21.56	24.71	24.60	25.00	22.46
Female	16.46	26.00	24.50	19.40	27.50	20.25
GIST		23.00	-	19.00	27.00	21.40
IDA		22.00	24.00	13.00	29.00	22.67
GLASS		22.25	20.00	23.00	27.00	22.40
FATA		27.00	25.67	27.00	24.20	25.50
Total	18.97	22.92	24.67	22.00	25.50	21.72

Table 3: Table displaying the means for each factor being analysed

in table 4. Due to the unbalanced nature of the data, sum of squares Type II was used.

Source	DF	SS	MS	F	*p*
Academic Level	4	429.534	107.383	3.893	0.007
Gender	1	39.507	39.507	1.432	0.236
Academic Level*Gender	4	202.473	50.618	1.835	0.133
Error	62	1710.410	27.587		
Corrected Total	71	2420.444			

Table 4: 2-way ANOVA significance and interaction between factors (note that in this instance, SS denotes Sum of Squares)

As can be seen here, the main effect of academic level is significant ($p<0.01$) whereas no significant effect of gender or interaction between the two factors was detected.

Once the analysis of variance identified that the main effect was significant, the effect size between groups was calculated using Hedges' g, favoured in this case over Cohen's D due to the small sample sizes of some groups. The results of this analysis are displayed in table 5.

	Level 1	H Level	MSci	PhD	Staff
Level 1	-				
H Level	0.672	-			
MSci	0.963	0.384	-		
PhD	0.480	-0.168	-0.487	-	
Staff	1.124	0.592	0.210	0.667	-

Table 5: Table displaying the effect size between groups, using Hedges' g

While recommending caution, Cohen suggests the following bounds can be considered for effect size:

- 0.2 as a small effect
- 0.5 as a moderate effect
- 0.8 as a large effect

Negative values indicate an effect in the inverse direction.

5.3 Discussion

As expected, with the exception of the PhD students, the average SS ability of each cohort increased as academic attainment increased. By examining the effect size between groups, we can see the Honours cohort being better than the level 1 cohort, the MSci cohort were better than them and so on. Although the incremental effect sizes are quite small, they compound to display large differences between cohorts on either ends of the scale.

A possible theory for the PhD students not fitting this pattern is that their backgrounds are considerably more varied than any of the other cohorts tested. While the first year students will have graduated from potentially differing high school programmes and curricula, all the students tested from the level 1 course specifically chose this course as they had limited programming experience. This balances the background of the cohort significantly. Honours and MSci students will most likely have undertaken different modules and will have different preferences and specialisations, but will all be some way along the same course at the same level of assessment. Staff members will also have had a varied background, however it can safely be assumed that they have achieved a relevant PhD and will have several years of experience.

Conversely, PhD students attending the University of Glasgow come from a wide range of degrees undertaken at universities around the world. Each of these courses will have different focuses, different teaching styles and different methods of assessment. As a result, it is possible that the way attainment and progression are measured for the current PhD students is very different to previous methods they have been exposed to. It must also be noted that the deviation in scores for the PhD students is high, and in fact the highest recorded score on the test (a perfect score of 30) was achieved by a PhD student - this seems to indicate that rather than the group generally having poor spatial skills, the spread is just much broader than in other cohorts.

With the exception of the anomalous PhD students, a clear pattern can be seen in that the average SS of those with higher levels of attainment are higher than those with a lower level of attainment. While this takes us a step closer to understanding the relationship between SS and CS, there are multiple conclusions which can be drawn from this result. One is that as one progresses in computing science, their SS are improved by the exercises and practices they are required to develop, as Pallrand noticed in physics. An alternative theory is that as cohort members progressed, only those with higher SS advanced, either because those with lower skills *could not* or *chose not to*. Both options are possible, and a longitudinal study of a cohort progressing through the system would be required to decidedly identify which hypothesis is true, if either.

Concerning the research section of each participant and the mean scores of these groups, the results partially support the study by Jones and Burnett, as the GIST section has the lowest average SS. However, if programming were the primary factor to which SS contributed as they suggest, it would be expected that the GLASS group would have the highest SS, since this is the section most focused on engaging with programs. However, the FATA section have the highest average SS, indicating that some other factor in computing is more likely to be closely related to SS.

This result supports the model proposed in this paper:

rather than programming itself being the core factor to which SS are contributory, we believe it to be the more abstract regions of mental model building, abstraction and problem solving, which extend beyond programming itself. These results support this hypothesis, indicating that the highest average SS can be found in a group whose primary focus is tackling theoretical, abstract problems rather than those who write the most programs.

6. SS TRAINING PILOT STUDY

With the knowledge that a connection between SS and CS exists, a theoretical model for *why* this relationship exists and the proof that as CS attainment increases so do spatial skills, the next major step would be to attempt to prove causality. That is, by increasing one's spatial skills, will we also see an improvement in their computing ability?

In order to test this hypothesis, a pilot study was run in the school of computer science with the CS1CT students (entry level students with limited programming experience, the same group who were who were a part of the previous experiment) which consisted of a spatial skills training course to be run alongside their degree. The experiment was designed in order to answer the following three questions:

- **RQ3:** Can spatial skills be effectively trained alongside a standard CS course as they can in engineering?
- **RQ4:** Will an improvement in spatial skills lead to an improvement in computing grades?
- **RQ5:** What are the practical aspects which must be considered in the delivery of such a course?

6.1 Course Materials and Delivery

The content for the spatial skills course was drawn from Sorby's workbook [35]. Many of the exercises in the workbook consist of multiple choice or short answer questions - these were put online on the institution's Moodle site in a dedicated course, and could be attempted by students at their convenience. Drawing exercises were uploaded as PDF files online which could be viewed by students and completed on drawing paper provided - depending on the exercises, squared paper and isometric grid paper were provided in the labs.

The exercises consist predominantly of exercises in training spatial visualisation and relations. The modules (chapters) delivered consisted of the following tasks:

- Modules 1 and 2 consisted of reading and understanding orthographic and coded drawings, followed by drawing orthographic and isometric designs (an example is shown in figure 10)
- Module 3 consisted of flat-pattern design and understanding - for example, given a flat pattern, deciding what 3D object could be formed
- Modules 4 and 5 consisted of rotations of objects and patterns about axes (similar to the test objects in the PSVT:R - an example is shown in figure 11)
- Module 6 consisted of reflections and symmetry
- Module 7 consisted of cutting planes and cross sections
- Module 8 consisted of revolutions of solids - for example, given a 2D pattern and an axis of revolution, identify the solid which would be produced
- Module 9 consisted of combining solids, involving extrusions, intersections, etc.

The course consisted of four one hour long sessions delivered over four weeks in the computing science labs where computers were readily available to access content. Students were advised to complete the online quizzes before attending the labs, then attempt the more complex drawing exercises in the lab under supervision.

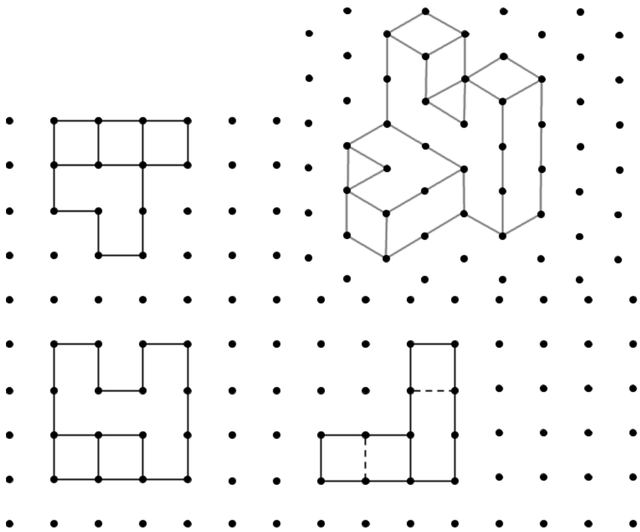

Figure 10: Exercise from Sorby's workbook on sketching isometric drawings from orthographic plans (with the correct answer drawn in grey)

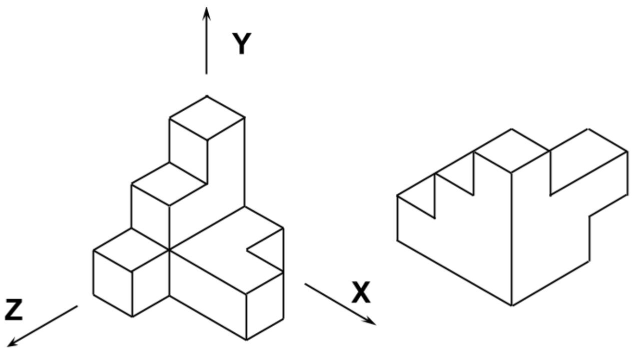

Figure 11: Exercise from Sorby's workbook on 3D rotations - given the 3D shape and axes displayed on the left, the student is required to identify the axes and degrees of rotation required to get the orientation on the right

It is acknowledged that the course was delivered on a much shorter timescale than Sorby would typically deliver the course (ten weeks) but aside from this being unavoidable due to time constraints, this also gave a perspective on students' attitudes to attempting exercises at home and the level of self-efficacy involved in the implementation of a course like this.

It must be noted that Sorby's workbook is primarily focused on the development of spatial visualisation and relations. Based on the model presented in this paper, it is believed that visualisation plays a significant role in computing aptitude, however other spatial skills factors are likely to contribute as well (namely closure flexibility, closure speed and perceptual speed) in the identification of cues. Ideally a fully fledged spatial skills training course would involve the testing and training of these factors as well. However it must also be noted that in all previous studies of spatial skills and computing, these factors have not been measured, so correlation has not been established. This is not to say that these factors are insignificant (the presented model suggests that they play a role in program generation and comprehension), but it is completely unknown what the effect of training and testing these factors will be.

6.2 Course Participation and Commitment

Every student in the CS1CT course was asked to take the PSVT:R during a lecture - this lecture was immediately after a class test, so attendance was very high. An invitation to the course was offered to every student who failed or marginally passed the PSVT:R (scoring 18% or less constitutes a fail, and scoring between 19% and 21% inclusively constitutes a marginal pass).

Of the 54 students invited to take part in the course, 18 opted to take part. Of these 18 students, 4 students attended every training session and fewer than half attended more than two sessions. Moodle allowed the online quizzes to be tracked, and the number of attempts at these was generally low, with few students completing them. Furthermore, only 5 students who had participated in the course attended a discussion session after the course had concluded to provide feedback and retake the PSVT:R. This level of commitment was very low, especially since the recommended level of commitment was only two hours a week with one of these hours being a lab session, with weekly emails reminding the students of the sessions and pointing to existing research suggesting why the training may be of benefit.

6.3 Results and Discussion

6.3.1 Improvement in Spatial Skills

To answer the first research question is a simple case of reviewing the pre- and post-test scores of those who undertook the training course. Unfortunately, with only five students attending the post-test session, it is hard to rely on these results, so **RQ3** cannot be reliably answered. They are displayed in figure 12. With the exception of one student, the rest showed substantial gains in their spatial skills after having taken the course.

6.3.2 Improvement in Computing

To measure if spatial skills training had any correlation with improvement in CS ability, the change in class ranking on each side of the SS training was determined for students in both experimental and control groups.

In order to do this, students were ranked at two milestones: first a class test, taken halfway through the first semester, and secondly their final grade at the end of the first semester which was calculated after the period of study

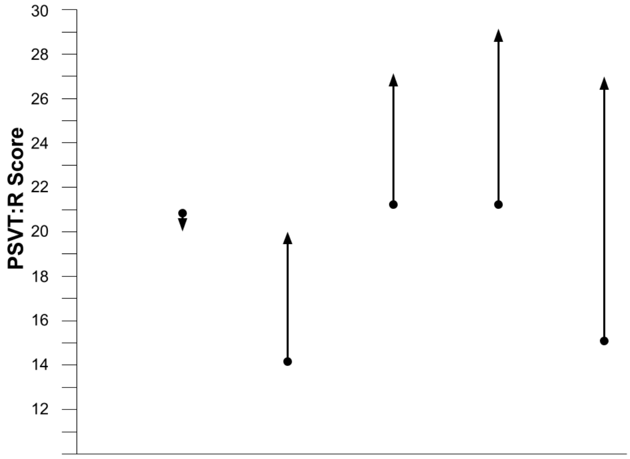

Figure 12: Chart displaying the improvement of students who took the spatial skills post test

was completed.

To remove a potential confound, only students intending computer science or similar were examined in this experiment. The reasoning behind this is that most students will take three courses in their first year, meaning that success in one of them could be a second or third priority in favour of their other courses. Given that this experiment reviews data related to computing aptitude, it could be misleading if students who are not fully committed to doing well were included. While these students were offered equal opportunities in taking the course, their final results were not included in the analysis of the data.

Students were divided into two groups: those who attended at least one training session (experimental) and those who did not take any training (control). After removing non-intending honours students, there were 8 students in the experimental group and 34 in the control group. The data was analysed with Welch's t-test. The groups were found to be significantly different ($p<0.05$, $t>t$-crit) and table 6 displays a summary of the results. The Cohen's d effect size was also calculated and displayed in this table - recall that with Cohen's d, an effect size of greater than 0.8 is considered large.

	Control	Experimental
Mean	-1.74	20.63
SD	29.70	15.86
n	34	8
t	**2.95**	
***t*-crit**	**1.73**	
Cohen's *d*	**0.94**	

Table 6: Table summarising the results of the t-test investigating the experimental and control groups in the spatial skills training course - the mean and SD values refer to the difference in ranking of the students in each group

This table does not reflect an interesting observation made when examining the data: the standard deviation is high in the control group, with a broad range of increases and decreases, however *every* student on the spatial skills course who had computer science as a major showed an improvement in the rankings or remained the same, with none of them dropping.

While this experiment indicates some promising results, it is important to remain cautious. As stated throughout the description of this course, student uptake was low and therefore these results may be misleading due to the small sample sizes. In addition to this, similar to Veurink and Sorby's study, self-efficacy cannot be ruled out as a factor. Given that the course was voluntary, it may only have been taken by those with a strong desire to improve (possibly related to growth mindset), meaning that spatial skills may not have been the only factor in their improvement. This is an argument for making this course mandatory in future experiments to rule out this factor and reduce the effect of confounding variables.

Based on this, it is hard to confidently answer **RQ4**. I do not feel that this experiment has concretely proven causality, however the results are promising and indicate that further research is necessary.

6.4 Post-Course Discussion Session

After the course had been completed, a discussion session was held with some of the course attendees to talk over the course and allow them to provide feedback. Ultimately, this feedback session was a very valuable takeaway from the pilot study - given the low uptake of the course it is hard to say for certain how reliable the statistical results are, but the qualitative feedback from the students taking the course gives some very useful insight on improving the course.

One of the first questions addressed was why the attending students thought that the attendance and quiz attempts were so low. The students suggested a combination of factors:

- Students did not have time to attempt quizzes outside of lab hours
- The optional nature of the course made students less likely to feel the need to attend sessions
- Students did not really understand what connection this course would have for them, or what benefit it might have in their degree

This appears to be in part a failure in the advertisement of the course. Students were given a brief introduction to the course in a lecture, a rundown of the benefits in the invitation email and a related paper was uploaded to the Moodle, however it is possible that these failed to properly "grab" the attention of the students and encourage interaction. Regardless, I believe that the best course of action to ensure the most reliable results in a full scale study would be to make the course compulsory for participants. As previously stated, this rules out self-efficacy as a factor and ensures a balanced level of interaction from all participants, limiting the number of factors which could confound the results.

Additionally, the course should really be run over a longer period of time, the ten weeks that Sorby intended. Not only does this reduce the pressure on the students in the sessions, but also reduces the need for exercises to be completed at home.

Students were also asked what they thought about the connection between spatial skills and computing science. Multiple students related the difficulty in performing complex mental operations (particularly the kind shown in figure 11 from module 5) to trying to solve a hard programming problem. To paraphrase one student:

> "It hurts in the same part of your head as when you're trying to work out what a big bit of code is doing."

Another student stated that they needed absolute focus for the more challenging exercises, which is also their preferred environment for programming. And finally, one made a comment about how being able to "see" rotations is like being able to "see" what a program does - once again this is similar to Sorva's comment concerning the ability to "envision with the mind's eye."

These connections are very surface level, particularly compared with the model presented earlier in this paper, but it is still interesting to see that even entry level CS students can identify some similarity in the underlying processes involved in both SS and CS. For future research, it would be sensible to have these testimonies recorded and used to convey the potential impact of such a course to try and maximise interest in doubtful participants.

Another statement from a few of the students was the accessibility of the course. One student suggested that the course be redesigned to be entirely homework based with drop-in sessions, indicating that the majority of the work could be self-learned. This statement was not met with universal agreement from the other students, however. Two indicated that the instructions provided - pages from Sorby's workbook introducing each module - were not quite enough to fully gain an understanding of what was required. One student suggested creating short video tutorials to help with the explanations. What was agreed, however, was that regardless of how much effort it took to initially get started with the work, the exercises were easy to pick up and perform, even for those who had not done anything similar.

It should also be stated that the overseer in the lab sessions has had high school level training in graphics and drawings similar to those in the exercises by Sorby, but no further, and was able to understand the exercises very quickly and aid the students in getting to grips with them. It is felt that this reflects the highly accessible nature of the course and its ease of use in the setting provided.

Finally, despite finding many of the exercises challenging, there was a universal attitude among the participants that was not expected: it was fun. This particularly applied to the drawings; the Moodle exercises were considered manageable but sometimes tiresome, but the students commented that hand sketching small diagrams was very enjoyable. In describing them, the following words were used:

> **fun, enjoyable, cathartic, engaging, intriguing, a bit different, relaxing**

While these are not universally applicable to all the exercises (some students made it clear that while they did enjoy the challenge of the multiple-axes 3D rotations, they would not describe them as *relaxing*) they were generally felt about most of the drawing exercises in some way. This indicates that this course carries a hook with which to draw in students to engage more, thus improving their willingness to participate and commit to the exercises. In addition, it provides an enjoyable and comfortable environment which is also likely to improve the students' ability in one way or another, which cannot be a bad thing.

6.5 Advisement on Future Courses

To conclude the discussion of this experiment, I shall summarise the findings concerning the practicalities of running such a course, be it as a dedicated training course or as a basis for further investigations. It is recommended to follow the following guidelines:

- **The course should be compulsory.** This is particularly important to researchers intending to investigate the effects of spatial skills training - to a degree it will rule out self-efficacy as a factor in improvement. Furthermore, in the context of training, it will ensure that the course is of benefit to those who wish to improve but do not see the connection between SS training and computing science.

- **Explain the reasons behind the course thoroughly.** This will also help remedy issues pertaining to the latter point in the first recommendation. Students who do not understand why they are taking the course are less likely to invest time, so making the connection clear is likely to be beneficial to them.

- **Spread the course out.** The constraints of this pilot study meant that the course needed to be conducted only over four weeks, which was noted by the participants as not enough time to cover the content. It may be tempting to condense the course to deliver it more quickly, but for more reliable results Sorby's guideline of ten weeks should be used.

- **Limit homework content.** Using an online platform was fine for this course, however students were not likely to invest much time in the course at home in their own time, and preferred to complete exercises in the lab. This relates to the recommendation above: spreading out the course and completing as much of the content in the lab as possible is advised.

- **Use testimonials and comments from previous attendees.** It was felt that if some of the students invited had been informed that the course was described as "fun" or "cathartic" by their peers, they would have been more likely to attend the course. By including some comments from other students about the enjoyability of the course, it is more likely that fence-sitting students will be swayed into looking more closely at the course.

Following these guidelines should maximise the efficiency and enjoyability of the course for the students, and relay some important findings made in this study.

7. CONCLUSION

In this paper I have reviewed literature concerning the relationship between SS and STEM, particularly in CS. This literature indicates that a correlation between SS and CS exists, with one study displaying what has been interpreted as a causal effect. Furthermore, I have identified that in one STEM field SS improve over a period of learning - not as much as if they had received directed SS training, but more than a liberal arts student - which indicates that the relationship is likely to be a biased two way relationship.

I have also collated and presented a substantial discussion of spatial skills themselves, condensing and summarising a broad field in a format which is easy to grasp for the relatively uninitiated. Based on this, I have presented a model for the relationship between SS and CS. This model is rooted in existing research into cognition in CS, particularly in program comprehension, program generation and problem comprehension. The model indicates that particular factors of SS likely have an effect in the reading and identification of key points in code or problems, as well as the mental models constructed in attempting to understand programs and theoretical problems.

I conducted an experiment to strengthen understanding of the relationship between SS and CS achievement, showing that in general the average SS of a cohort increases with academic attainment, which extends the research undertaken by Jones and Burnett. The experiment also supports the model connecting SS with CS, as the participant research section with the highest average SS was the section who engage in the most abstract and theoretical thinking.

And finally I have presented the findings of a pilot study attempting to train spatial skills and investigate the effect this has on computing ability. Although the results of this study are marred by the low level of commitment to the course, they are largely encouraging, and the discussion of the practical aspects of running such a course will be helpful in laying the groundwork for future experiments.

My contribution furthers our understanding of SS and their relationship with CS and lays the foundation for a larger experiment to determine if the relationship is causal. If SS training does benefit computing ability substantially, then it is worth introducing on a large scale, due to its cost-effectiveness, high accessibility and easy implementation.

8. REFERENCES

[1] L. J. Burton and G. J. Fogarty. The factor structure of visual imagery and spatial abilities. *Intelligence*, 31(3):289–318, 2003.

[2] J. B. Carroll. *Human cognitive abilities: A survey of factor-analytic studies.* Cambridge University Press, 1993.

[3] C. S. Carter, M. A. Larussa, and G. M. Bodner. A study of two measures of spatial ability as predictors of success in different levels of general chemistry. *Journal of research in science teaching*, 24(7):645–657, 1987.

[4] CEEB. Ceeb special aptitude test in spatial relations, developed by the college entrance examination board, usa, 1939.

[5] S. L. Coleman and A. J. Gotch. Spatial perception skills of chemistry students. *Journal of Chemical Education*, 75(2):206, 1998.

[6] S. Cooper, K. Wang, M. Israni, and S. Sorby. Spatial skills training in introductory computing. In *Proceedings of the eleventh annual International Conference on International Computing Education Research*, pages 13–20. ACM, 2015.

[7] F. Détienne and E. Soloway. An empirically-derived control structure for the process of program understanding. *International Journal of Man-Machine Studies*, 33(3):323–342, 1990.

[8] B. Du Boulay. Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1):57–73, 1986.

[9] R. B. Ekstrom, D. Dermen, and H. H. Harman. *Manual for kit of factor-referenced cognitive tests*, volume 102. Educational Testing Service Princeton, NJ, 1976.

[10] E. Fennema and J. Sherman. Sex-related differences in mathematics achievement, spatial visualization and affective factors. *American educational research journal*, 14(1):51–71, 1977.

[11] S. Fincher, B. Baker, I. Box, Q. Cutts, M. de Raadt, P. Haden, J. Hamer, M. Hamilton, R. Lister, and M. Petre. Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses. 2005.

[12] J. W. French. The west point tryout of the guidance battery. *ETS Research Report Series*, 1951(1), 1951.

[13] J. W. French. The west point tryout of the guidance battery, part 2. *ETS Research Report Series*, 1955(1), 1955.

[14] R. Guay. *Purdue Spatial Vizualization Test.* Educational testing service, 1976.

[15] R. R. Hake. Relationship of individual student normalized learning gains in mechanics with gender, high-school physics, and pretest scores on mathematics and spatial visualization. In *Physics education research conference*, number 2, pages 30–45, 2002.

[16] J. R. Hills. The relationship between certain factor analyzed abilities and success in college mathematics. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES PSYCHOLOGICAL LAB, 1955.

[17] C.-H. Ho, C. Eastman, and R. Catrambone. An investigation of 2d and 3d spatial and mathematical abilities. *Design Studies*, 27(4):505–524, 2006.

[18] S. Jones and G. Burnett. Spatial ability and learning to program. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 2008.

[19] M. Kozhevnikov, M. A. Motes, and M. Hegarty. Spatial visualization in physics problem solving. *Cognitive Science*, 31(4):549–579, 2007.

[20] M. Mandell. Selecting chemists for the federal government. *Personnel Psychology*, 3(1):53–56, 1950.

[21] M. M. Mandell and S. Adams. Selection of physical scientists. *Educational and psychological measurement*, 8(4):575–581, 1948.

[22] M. G. McGee. Human spatial abilities: Psychometric studies and environmental, genetic, hormonal, and neurological influences. *Psychological bulletin*, 86(5):889, 1979.

[23] D. I. Miller and D. F. Halpern. Can spatial training improve long-term outcomes for gifted stem undergraduates? *Learning and Individual Differences*, 26:141–152, 2013.

[24] D. A. Norman. Some observations on mental models. In *Mental models*, pages 15–22. Psychology Press, 2014.

[25] G. J. Pallrand and F. Seeber. Spatial ability and achievement in introductory physics. *Journal of Research in Science Teaching*, 21(5):507–516, 1984.

[26] J. R. Pribyl and G. M. Bodner. Spatial ability and its role in organic chemistry: A study of four organic courses. *Journal of research in science teaching*, 24(3):229–240, 1987.

[27] R. S. Rist. Schema creation in programming. *Cognitive Science*, 13(3):389–414, 1989.

[28] R. S. Rist. Program structure and design. *Cognitive Science*, 19(4):507–561, 1995.

[29] A. Robins, J. Rountree, and N. Rountree. Learning and teaching programming: A review and discussion. *Computer science education*, 13(2):137–172, 2003.

[30] C. Schulte. Block model: an educational model of program comprehension as a tool for a scholarly approach to teaching. In *Proceedings of the Fourth international Workshop on Computing Education Research*, pages 149–160. ACM, 2008.

[31] S. A. Sorby. Developing 3-d spatial visualization skills. *Engineering Design Graphics Journal*, 63(2), 1999.

[32] S. A. Sorby. Developing 3d spatial skills for engineering students. *Australasian Journal of Engineering Education*, 13(1):1–11, 2007.

[33] S. A. Sorby. Educational research in developing 3-d spatial skills for engineering students. *International Journal of Science Education*, 31(3):459–480, 2009.

[34] S. A. Sorby and B. J. Baartmans. A course for the development of 3-d spatial visualization skills. *Engineering Design Graphics Journal*, 60(1):13–20, 1996.

[35] S. A. Sorby and A. F. Wysocki. *Introduction to 3D Spatial Visualization: an active approach.* Cengage Learning, 2003.

[36] J. Sorva. Notional machines and introductory programming education. *ACM Transactions on Computing Education (TOCE)*, 13(2):8, 2013.

[37] R. F. Street. A gestalt completion test. *Teachers College Contributions to Education*, 1931.

[38] D. E. Super and P. B. Bachrach. Scientific careers and vocational development theory: A review, a critique and some recommendations. 1957.

[39] L. A. Tartre. Spatial orientation skill and mathematical problem solving. *Journal for Research in Mathematics Education*, pages 216–229, 1990.

[40] L. A. Tartre. Spatial skills, gender and mathematics. In E. Fennema and G. Leder, editors, *Mathematics and Gender*, chapter 3, pages 27–59. Teacher's College Press, New York, 1990.

[41] D. M. Tracy. Toys, spatial ability, and science and mathematics achievement: Are they related? *Sex roles*, 17(3-4):115–138, 1987.

[42] D. H. Uttal, N. G. Meadow, E. Tipton, L. L. Hand, A. R. Alden, C. Warren, and N. S. Newcombe. The malleability of spatial skills: A meta-analysis of training studies., 2013.

[43] N. L. Veurink and S. A. Sorby. Raising the bar? longitudinal study to determine which students would most benefit from spatial training. In *American Society for Engineering Education*. American Society for Engineering Education, 2011.

[44] J. Wai, D. Lubinski, and C. P. Benbow. Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance.

Journal of Educational Psychology, 101(4):817, 2009.

[45] L. Wise, D. McLaughlin, and L. Steel. The project talent data handbook, revised. *Palo Alto, CA: American Institutes for Research*, 1979.

[46] M. A. Wittig, S. H. Sasse, and J. Giacomi. Predictive validity of five cognitive skills tests among women receiving engineering training. *Journal of Research in Science Teaching*, 21(5):537–546, 1984.

[47] S. Y. Yoon. *Psychometric properties of the revised purdue spatial visualization tests: visualization of rotations (The Revised PSVT: R)*. Purdue University, 2011.