Project Proposal

Legend:
Original Proposal
Updates from Meeting 1 [May 31st, 2018; 6PM - 7PM]

Big Picture

What is the overall problem that this and related research is trying to solve?
Why should people care about the problem?
What is the general approach to solving this problem?
How will this approach result in a solution?
What is the value of this approach beyond this specific solution?

Create and include one or two graphics that capture and communicate the problem and proposed solution to technical but non-expert audiences. Can you create a one or two sentence summary of the problem and the proposed solution approach?

Traditional computer-aided design (CAD) software continues to hinder casual end-users when fabricating designs, as it offers an unrealistic design environment that overwhelms users with prerequisite learning, and tasks them to design in a virtual environment through a 2 dimensional user interface. To enhance support for furniture design in CAD technology, this project proposes a software that offers a 3D collaborative design environment and utilizes virtual reality (VR) to transform the current CAD process into an accessible design experience for furniture designers alike.

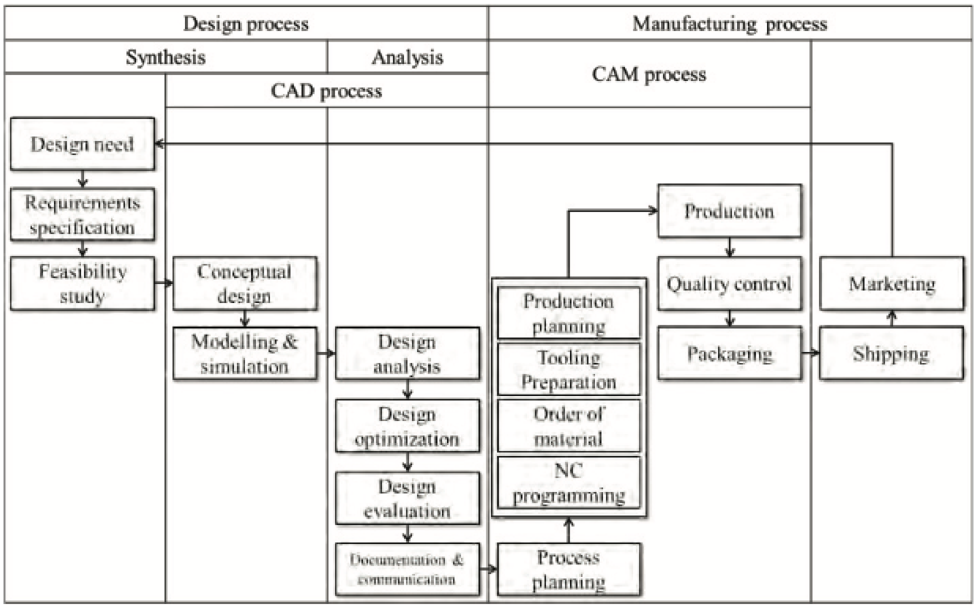


Figure 1. Overview of the role of CAD in design and manufacturing processes. Replicated from [7].

While CAD software has traditionally offered opportunities for designers to synthesize, create and share manufacturable solutions for persisting design problems, the introduction of virtual reality and gesture detection can better enhance the design process. As illustrated in Figure 1, CAD software operates by modeling a conceptual design using predefined design specifications and, using built in libraries and toolkits, optimizing the design for manufacture. User studies from [6, 7, 10, 11, 12] evidenced that popular CAD software were originally designed without consideration of its usability, employing a traditional 2-dimensional graphic user interface (GUI) that constrains users to design in a virtual environment isolated from the 3-dimensional setting in which the design is physically implemented [5]. Although GUIs offer better I/O performance and are considered the standard for human-computer interaction, in order for users to generate a 3D object through a 2D device screen, CAD software takes input from users through a mouse-and-keyboard system, which is less familiar to users who conventionally design through sketching and other physical activities [6, 14]. In addition, CAD softwares have resorted to software-specific libraries that allow users to navigate and design through a 2 dimensional interface, which demand extensive understanding of manufacture specifications, shape manipulation and software-specific design tools before users can create their intended design [7, 10, 11, 12]. The excessive domain-specific knowledge required to model the design problem on CAD software, in conjunction with the steep learning curves required to utilize predefined libraries and tools have resulted in reduced productivity and unsatisfactory design experiences [7].

In contrast, user experiments have demonstrated that gesture-based interfaces (GBI) are more accessible to users of different skill levels, as they offer better hedonic experiences, are easily learned, and leverage existing motor and dexterous skills to design on a computer [2, 6, 14]. In particular, with the introduction of virtual 3D environments, particularly virtual reality (VR), such gestures can be detected within the physical 3D environment and transform into design fabrications [2, 6]. While research investigating the usability of virtual reality in CAD have been limited, products incorporating VR have increased in the past decade, motivated by its benefits to human-computer interactions. VR softwares are also increasingly portable across mobile and desktop devices, which can offer enhanced design experiences for users beyond the desktop environment. [1, 3, 14] have agreed that Virtual Reality (VR) offers a better insight of 3D objects by providing direct drawing and editing through 3D interaction media, offering greater creative freedom "through linking creative experimentation and accurate manufacturing-oriented modelling." For designers, whose end goal is to implement their designs in context to its physical setting, gesture detection in 3D virtual environments can bridge the void between fabricating and manufacturing a design, simplifying the design process and improving design productivity.

Update: [31/05/2018]

After further research, we propose that as opposed to VR environments, augmented reality (AR) environments may better achieve a design solution that can aid designers when fabricating design in relation to their physical environments. As documented in [25, 26, 27], the implementation of 3D modelling in AR is a new field, and has only been made possible due to advancements in VR and camera technology. However, with demands to physically simulate and manufacture prototypes for products, which is imperative to the design evaluation and optimization process, designers can easily communicate ideas through AR based modelling, as it is based in the physical environment, detected by the camera. Because VR and AR share many overlaps in research, function, and implementation, the benefits of VR still hold for AR.

In contrast to VR, however, which immerses users into a 3D environment isolated from the physical world, "AR offers a more realistic experience because the user interacts with real objects, while VR manipulates virtual objects and is limited by the lack of suitable sensors feedback. Further, VR is oriented to training rather than guidance, improving the skills of users through multiple simulations, while AR can be used for both training and guidance." AR, in conjunction to the aforementioned benefits of gesture detection in CAD, can enhance user's hedonic experience and ease of learning, as user's gestures can be detected in real time using a camera, in contrast to a remote controller, which is typically used with VR to translate physical gestures into virtual interactions. [30, 31, 32, 33] have indicated the portability of AR in everyday devices such as a smartphone, which makes AR accessible beyond the indoor, desktop-oriented environment that VR typically offers. Designers choosing to design on a mobile device, or design outdoors, can employ such features to design outside of the typical desktop environment.

In relation to 3D environments, research has underscored the importance of collaborative utilities in producing complex solutions and improving productivity in design practices and manufacturing [19, 20, 23]. Typically, CAD has been performed in isolation, where designers work in their independent environments, and communicate through transfering files in the form of email, video, or other conventional methods. Recent advances in both VR and collaborative virtual environments (CVEs) have offered opportunities for users to interact not only with virtual objects in a 3D environment, but also with other users in real time. The applications of CVEs have been demonstrated in areas such as education through 3D environments [18], remote social interaction [19], and virtual tours and presentations of virtual objects [20]. The ability to collaborate with peers and model in 3D space can greatly benefit designers, who typically work in teams, communicating with manufacturers, clients, and affiliated parties to produce their product. It also benefits casual users, who may wish to learn CAD under the guidance and support of peers and mentors. In addition, presenting designs in a 3D CVE can potentially save costs required to otherwise prototype a physical model. Though such collaborative environments have been explored in virtual reality (VR) consoles, they have rarely been used in conjunction with gesture-based interfaces for CAD, which leads us to the crux of this project.

Update: [31/05/2018]
The added benefit of collaborative spaces using AR can also aid designers when personalizing the product for their targeted clients. Products, particularly furniture, are frequently design in context of the person's space and environment. By offering a real-time collaborative space where both the users of the product and the designer can cooperate to produce the desired product meeting the user's satisfaction. Existing products offer limited collaborative features beyond email, sharing files, and adding comments after saving files, limited users to asynchronous, distributed interactions. However, AR, which not only allows remote collaboration, but also visualization of the physical space, can offer opportunities for asynchronous distributed interaction.

Specific Project Scope

What subset of the overall problem are you addressing in particular?
How does solving this subproblem lead towards solving the big picture problem?
What is your specific approach to solving this subproblem?
How will we know that this subproblem has been satisfactorily solved?
What is the value of your solution beyond solely solving this subproblem?

The aim of this project is to develop a sketching software that combines gestural movements with conventional digital fabrication methods and CVE for CAD in a collaborative 3D environment. Specifically, the focus is to simplify the process of furniture fabrication by letting casual end-users collaborate in real time through a VR environment, to design and produce manufacturable furniture objects.

The product will utilize an existing backend software that optimizes furniture-specific design processes, alleviating designers from the engineering specifications required during computer-aided manufacturing (CAM). Although research in CAD have investigated possible implementations of CVEs and VR, most have only looked at these two areas in isolation from each other. With the needs to design and collaborate in a realistic environment representative of the physical world, research must consider CVEs and VR in tandem.

Update: [31/05/2018]
In relation to the needs of fabricating designs directly in the physical world, such that designers can manufacture furniture relative to the aesthetics and ergonomics of the surrounding design space, we believe that virtual reality (VR) is not the most effective solution. Although it allows users to develop models in 3D environments and incorporate more natural hedonically pleasing gestures such as sketching, it still requires users to design in a virtual environment isolated from the physical space in which the design is implemented. To address this issue, we propose the benefits of using Augmented Reality, which tracks physical surfaces and bodies in the environment using camera-based hardware.

In addition to visualizing a CAD fabrication in the physical environment, the ability to evaluate this virtual product by simulating tests to examine its functions is tantamount to the product's physical implementation. With libraries to simulate physics in AR, the proposed software should also offer virtual simulations that demonstrate the product's functions, such as its durability, weight, stability, etc.

Though CAD technology has been utilized in numerous contexts (e.g. architecture, graphics design, interior design, etc.), this project will focus specifically on producing a sketching software that aids furniture design. Furniture, which incorporate human ergonomics specifications and must be designed in consideration of its surrounding aesthetics, tax designers with knowledge pertaining not only the manufacturing specifications, but also client-based needs [21]. The function of the furniture and the space in which it is implemented are inextricable from each other, yet the needs to personalize the fabrication and optimize its design with traditional CAD software typically takes away from the actual design experience [21, 22]. Utilizing VR and CVE technology, furniture, which are typically placed in scope of controlled, indoor environments, can thereby be designed within this fixed virtual environment, and modelled to scale relative to its surroundings. In addition, utilizing VR in an indoor, closed environment allows users to conveniently model their ideas based on existing physical products, achieving precise and improved ergonomic solutions. Thus, it is believed that the opportunity to directly design furniture within the scope of its physical setting, using the aforementioned qualities offered by gesture detection and CVEs is particularly beneficial for furniture design processes.

Update:
The introduction of mobile AR also implements features to scan the physical environment and visualize dimensions of the real world environment. As furniture design emphasizes human ergonomics, dimension handling is imperative to the furniture's physical implementation. Instead of estimating its size, or physically measuring the environment with a ruler, AR implementations offer the ability to track real world dimensions of physical objects based on its relative distance from the camera [31, 32]. Doing so allows users to assess environmental constraints and fabricate a design simultaneously.

In scope of this problem, a user study will be conducted comparing the experience and performance of furniture design fabrication using conventional CAD software and the newly created product. As the software aims to resolve unsatisfactory user experiences in CAD, an enhanced user experience and ease of learning will demonstrate the effectiveness of this product in driving human computer interaction in CAD.

Related Work

What foundation and fundamentals need to be known in order to understand the your problem, approach, and solution?
What work has been done before on this specific problem?
What are related problems that have been addressed?
What work has been done on those related problems?
How does this past work contribute to your proposed solution?

Be sure to cite all potential sources, and summarize each one in terms of its content and relation to your project.

Prior work have investigated CAD using gesture interactions and collaborative 3D environments in isolation, but have failed to offer solutions that addresses both issues. This work is built upon prior examples referenced in [14, 15, 16], which studies sketching and gesture detection in 3D space, and [18, 19, 20], which investigates CVEs for CAD. We also draw from research pertaining to Virtual Reality software and consoles, as it offers a feasible GBI for 3D CVEs. The proposed solution will integrate ideas and working models established by innovations from aforementioned categories of work, to develop a cohesive system that offers both gesture detection and collaborative environments in 3D Space for CAD.

Gesture-Based Interfaces in 3D Space for CAD:

[14] 2D Monitor displays the given virtual object. Users control the design using a mouse that has force-feedback. Forces detected by the mouse are parameterized and output as a 'sketched curve' using ClayTools, a VR-based software application designed for being integrated with gesture detection and sensable haptic devices]. To detect the orientation of the sketched curve and the direction the mouse is moving in 3D space, the VR environment uses a dual-camera setup that is used to track 3 Dimensional movement.

[15] Spatial Sketch: Uses Wii-based Infrared sensors to detect arm movement in a 3D space and transform the embodied interaction into parameterized design specifications. The user is provided with an infrared pen light and sketches in front of a screen. Two infrared cameras and triangulation surrounds the screen to detect the intensity and motion of the pen light. The change in light depth and orientation is then translated into parameters that are reflected on the resulting virtual space. Similar 3D input based sketch systems include CavePainting (virtual sculpting environment), Surface Drawing

Sketch Furniture Project FRONT: Motion-capture system designer firstly sketches in physical space to create 3D geometry for furniture design. Manually processed in 3D studio MAX before outputted to a 3D printer for full-sized fabrication: https://www.youtube.com/watch?v=8zP1em1dg5k

[16] DreamSketch: A 3D design interface that combines the free-form and expressive qualities of sketching with the computational power of generative design algorithms. "In DreamSketch, a user coarsely defines the problem by sketching the design context. Then, a generative design algorithm produces multiple solutions that are augmented as 3D objects in the sketched context. The user can interact with the scene to navigate through the generated solutions."

[17] ILoveSketch: http://www.dgp.toronto.edu/~shbae/pdfs/Bae_Bala_Sing_2008_ILoveSketch.pdf

TiltBrush: An existing game-based software for VR allowing users to sketch and draw in 3D space. Users put on a VR headset and holds two controllers that are then. Typically, the right controller controls the virtual 'brush' that users utilize for designing, and the left controller is a rotatable menu designed for changing parameters of the brush such as brush size, brush stroke, stroke color, etc. Tiltbrush has been tested and implemented by Google, and an online Open-Source Library is available on GitHub for use. Tiltbrush can be implemented on Game development platforms such as Unity and Unreal Engine 4, which offers necessary portability for VR across multiple devices and platforms. Videos of TiltBrush can be found here: https://www.youtube.com/watch?v=4OF8lLvFP9s

Collaborative Utilities in 3D Space:

[13] Syco3D: Real Time Collaborative 3D CAD System. Similar utilities cited included "Shared 3D Viewer from Hewlett Packard24, OpenSpace from CoCreate25 and IRIX Annotator from Silicon

Graphics26, allow the collaborative review of 3D models created from standard CAD applications." Similar products include Co-CAD, which "supports real-time collaborative solid modelling for mechanical engineers."

Mobile CVE (Virtual Stiklestad): Offers virtual tours for Norway heritage museum used by both local and distant learning communities. The environment is developed using Active Worlds (AW) software development library and 3D Studio Max. It was later implemented for remote mobile users using a PDA Mobile Client, all of whom enter into the environment with a customized avatar that tracks his or her location in the environment.

Augmented Reality and 3D Environments:
Current marketable VR technology (e.g. HTC Vive, Oculus Rift, Wii Console, SteamVR, etc.) have utilized VR libraries and technology to offer gaming and interactive experiences utilizing user's gestures in a controlled, indoor environment. Such applications are directly produced from existing game development engines such as Unreal Engine 4 (UE4) and Unity, both of which offer tutorials and access to open source libraries for users to equip gesture detection into their software.

In combining these VR equipment with the physical environment, technologies and software development kits (SDK) have emerged for programmers and developers to design and capitalize on existing hardware to bring augmented reality to portable devices.

[26] An example of that is FurnitAR, which utilizes the camera features of existing tablets to develop a home interior design software, using premade FBX objects, users can easily upload and drag the furniture into the desired location. https://www.youtube.com/watch?v=D5OwqXmrHS8

Other Examples:

1. ARCore with Google TiltBrush: https://www.youtube.com/watch?v=8HYy5X6WUTE
2. ARKit with Tiltbrush-based Mechanisms: https://www.youtube.com/watch?v=bTHaEg-bnaU
3. ARKit Paint on Mobile: https://www.youtube.com/watch?v=Kj8mWHkBTTw&t=258s
4. ManoMotion SDK: https://www.youtube.com/watch?v=TtpCaV9foDI; https://vimeo.com/236170510 (With integration with ARKit)

Relevant Material and Utilities for Implementation

The implementation of augmented reality and gesture detection are separate from each other, as the software development kits (SDKs) are often implemented independently for general purpose audiences. To deploy a project with the capabilities of both augmented reality and gesture detection, more than 1 SDK libraries will be deployed. The following is a list of resources available for the possible SDKs for AR and Gesture Detection in 3D space.

Developmental Environments (IDE) for Software Deployment

Different applications have different methods limitations and available libraries for developing software deployable for different devices (e.g. XCode, Android Studio, Visual Studio, etc.). Specifically for AR Software and Gesture Recognition, Unity and UE4 offer the best support for AR implementation, as it offers the ability to build to multiple types of devices and has access to an array of libraries.

Unity Documentation: https://unity3d.com/unity/features/multiplatform/vr-ar
As most existing AR/VR libraries are accessible within Unity, it is the safest and most accessible choice. While this project will most likely focus on one specific type of implementation (e.g. iOS phones, VR gear), unlike other IDEs, the code in Unity don't have to be rewritten if the hardware changes, because it supports multiplatform, and users simply have to change the build target (i.e. Android, iOS, VR) to update the resulting code from the project.

Unreal Engine 4: https://www.unrealengine.com/en-US/vr
In a similar manner, UE4 functions in a similar manner as Unity, offering multiplatform support for VR/AR using one single implementation of code. However, unlike Unity, UE4 is relatively new into the VR/AR space, which is why, as shown on the existing documents, there is less documentation for the libraries and implementation in UE4 as opposed to Unity. As AR is still a new space, particularly for software development, documentation of libraries and ways it can be integrated into IDEs is crucial. Thus, while UE4 offers an alternative solution, it is at risk of less understanding of how it works.

Alternative IDEs include Android Studio and XCode. However, such implementations are either tailored towards iOS or Android. Documentation are still provided below:

Android Studio [limited to exclusive support of ARCore/Google VR deployed only for Android]:
https://developers.google.com/ar/develop/java/quickstart
https://developers.google.com/vr/develop/android/get-started

XCode [limited to exclusive support of ARKit/Google VR deployed only for iOS]:
https://developer.apple.com/arkit/
https://developers.google.com/vr/ios/

Augmented Reality Software Libraries

General Comparison: https://thinkmobiles.com/blog/best-ar-sdk-review/

Google AR SDK: ARCore https://developers.google.com/ar/develop/

- List of supported devices: https://developers.google.com/ar/discover/supported-devices
- Development for Unity: https://developers.google.com/ar/develop/unity/quickstart-android
- Development for Unreal: https://developers.google.com/ar/develop/unreal/quickstart
- Development for Web: https://developers.google.com/ar/develop/web/quickstart
- iOS Compatibility: https://developers.google.com/ar/develop/unity/quickstart-ios

Recent Deployment of ARCore Version 1.2.:
https://blog.google/products/google-vr/experience-augmented-reality-together-new-updates-arcore/

Gesture Detection in ARCore [only for gestures on a 2D screen (not in 3D space)]:
https://developers.google.com/ar/reference/java/com/google/ar/sceneform/ux/GesturePointersUtility

Apple's AR SDK: ARKit

- Compatible Devices: https://developer.apple.com/documentation/arkit#overview
- Development for Unity: https://unity3d.com/learn/learn-arkit
- Development for UE4: https://www.unrealengine.com/en-US/blog/getting-started-with-ue4-and-arkit
- No Android Compatibility
- No Web support

Requirements for XCode: https://developer.apple.com/arkit/

EasyAR: https://www.easyar.com/

- Compatible Devices: https://www.easyar.com/view/sdk.html
- Development for Unity: https://www.easyar.com/doc/EasyAR%20SDK/Getting%20Started/2.0/Setting-up-EasyAR-Unity-SDK.html
- No Development for UE4
- No Web Support

ARToolKit [Open-Source Library]:
https://github.com/thinkmobiles/ARInvestigation-ARToolKitTest

- Compatible Devices: Android, iOS, Linux, Windows, Mac OS and Smart Glasses.
- Development for Unity: Previously supported via Asset Store but no documentation and asset was recently removed
- No Development for Ue4

- No Web Support

Vuforia: https://library.vuforia.com/ (Requires purchasing of license)
- Compatible Devices: Android, iOS,
- Compatible IDEs: Unity
- No Development for UE4
- No Web Support

Gesture Detection

Leap Motion: https://github.com/leapmotion/UnityModules/wiki
- Unity Documentation: https://developer.leapmotion.com/documentation/unity/namespaces.html
- Other Documentation: https://developer.leapmotion.com/documentation/index.html?proglang=current
- Offers custom gesture detection (e.g. pinching)
- Available for both VR and AR implementation

ManoMotion: https://www.manomotion.com/summary-sdk-developers/
- Documentation and SDK only provided after licensing
- Integrating with Apple's ARKit

ClayIO: http://clayair.io/
- Unknown documentation
- Unknown compatible IDEs: given an API bundle
- Compatible devices: iOS and Android (beta)

Goals, Deliverables, Tasks

Recursively break down the proposed project starting from the highest level specifications spanning a complete 1-2 term period down to individual atomic steps spanning days to at most a week. At each level of hierarchy, specify:
- *What goals do you need to achieve?*
  - *What are the specific questions that you will answer?*
  - *What skills / abilities will you enable?*
- *What deliverables will you produce to indicate that the above goals have been achieved?*
  - *These must be specific, concrete nouns.*
  - *How do these deliverables prove that the goals have been met?*
- *What tasks are necessary to generate those deliverables?*

- *These must include specific, concrete verbs.*
- *These often generate goals at the next lower level of hierarchy.*
- *Be sure to include dependencies --- which goals are necessary to have completed before starting each task?*

Distill the entire hierarchy into a list of weekly milestones. What will you need to achieve by when in order to attain your goal for the end of the project on time?

Goals

1. How can we detect design-specific gestures in a 3D virtual environment?
2. How can we transform these design-specific gestures into a virtual object that meets design specifications and manufacturing parameters?
3. How can we simulate the user's physical surroundings into the 3D virtual environment?
4. [If there is time] How can we track and detect real-time modifications to a design in the same 3D environment?
5. [If there is time] How can we use the existing capabilities of mobile and portable devices to visualize this virtual object/environment?

Deliverables

1. A working sketching software that meets the specifications of the project outlined in the project scope. Details are as follows:
    a. The software must offer measuring utilities to scan the physical design space, produce dimensions of the space, and visualize such dimensions for design implementation.
    b. The software must save, modify, and export design fabrications into design parameters and specifications that can be evaluated, tested, and optimized using a predefined back-end furniture compiler (refer to Wenzhong).
    c. The software must import existing design objects and fabrications into the physical design space (file format TBD, but typically .fbx, .obj, .stl files).
    d. The software must detect, track, and distinguish gestures (the number and type of gestures TBD) used to fabricate a design. Specific purposes of such gestures are as follows:
        i. Sketching the design
        ii. Editing the design (scale, translate, rotate)
        iii. Navigating the design space
        iv. Saving the implementation
        v. Exporting the implementation
        vi. Importing an existing implementation
        vii. [Any other particular features of gestures?]

    e. The software must offer virtual simulations used to test and interact with design fabrications.
2. A video showcasing the process of designing and implementing a physical manufacturable furniture (e.g. a chair) using the product in [1]. The product and example used for the video will be TBD.
3. [If Goal 4 is met] A video showcasing multiple users completing the task outlined in [2] with the product outlined in [1].
4. [If Goal 5 is met] A responsive user interface that visualizes the virtual object and environment produced using [1] on multiple Android and iOS devices (shown through screenshots and videos).

Tasks and Tentative Deadlines

Goal 1 [07/20]: Produce a working prototype of the product (meeting minimum requirements outlined in deliverables)

- Need not be perfect but at least roughly performs what it is supposed to do
- Slight, context-based issues can be fine-tuned in 2
- At least present 1 scenario where the product works and meets Goals

1. Compile a list of reference work, available (potential) open source libraries and necessary IDEs; compatible devices (e.g. Android watch/Apple watch to do sketching) and possible hardware
    a. Compile a list of viable Hardware implementations based on existing software and SDK/libraries for AR-based implementation, including costs, proprietary information, requirements, prerequisite components (e.g. licensing and certification)
    b. For each of the possible implementations, offer a general description of the procedures required to implement the required product outlined in the goal
    c. Evaluate limitations of the hardware/software implementation
2. [By or before end of Spring Quarter 06/15] Decide on hardware and software implementations
3. [By or before Summer Session 06/25] Using (2), offer a top-down system specification of the individual components/checklist to implement (to be pieced together).

Details from here on-forth are susceptible to updates and modifications based on (2) and (3)

4. Design an AR-based dimension specification system for scanning and visualizing dimensions of physical design space using hardware from (1).
5. Develop a working gesture-detection interface in 3D environment for sketching (working prototype) based on the dimension specifications outlined in (2).

6. Transform sketching gestures into parameters for a virtual object and manufacturing specifications/parameters; offer saving, editing, exporting, and importing features outlined in product deliverables.
7. Work with Wenzhong to transform established parameters into a manufacturable CAD file specification.
8. Implement save, edit, export and import features for software into manufacturable CAD files and 3D object ‘meshes’.
9. Implement gesture detections such as rotating, translating, scaling, and erasing utilities.
10. [If there is time, and if applicable] Implement multi-user design implementation for deliverables.
11. Video a version of the prototype for reference based on Deliverable [2].

Tentative Subgoals for Goal 1:

- 80% done by 7/13
- Demo to team by 7/20- receive feedback based on system

Goal 2 [08/20]: Debug, modify, and fine-tune the demo to incorporate and enhance UI/UX for casual users/designers

- Evaluate with team what deliverables/goals have (or have not) been met, and using that, update project
- Final product should be ready by 08/20
- Add subsequent tasks based on demo below

Tentative Subgoals for Goal 2:

- Ready for lab mates to test by 08/13
- Begin to plan and design UX experiment between Goal 1 and Goal 2.

Goal 3 [08/27]: Gather user experience data for product

References (to be updated)

[1] Kwon, J., Choi, H., Lee, J., & Chai, Y. (2005). Free-Hand Stroke based NURBS Surface for sketching and Deforming 3D Contents. Paper presented at the PCM 2005, Part I, LNCS 3767.

[2] Zheng, Chan & Gibson (2001)

~~[3] Teicholz E. CAD/CAM Hand Book Designing with Systems Series. New York: McGraw Hill, 1985~~

[4] ZAKRIA, GUAN, RIAZ, JAHANZAIB, & KHAN (2010). DOI 10.1007/s11465-010-0017-x https://link.springer.com/content/pdf/10.1007%2Fs11465-010-0017-x.pdf

[5] Kasik, Buxton, & Ferguson (2005). Ten CAD Challenges. IEEE Computer Graphics and Applications, 25(2), 81-92, IEEE Computer Society Press.

[6] Beurden, Wijnand A. Ijsselsteijn, and Yvonne A.W. de Kort (2012).

[7] Li, Hedlind, Kjellberg (2015). Usability Evaluation of CADCAM: State of the Art.

~~[8] Rasummen J. Skills, rules, and knowledge; signals, signs and symbols, and other distinctions in human performance models. Systems, man and cybernetics IEEE Transactions on 1983. 257-266. (3).~~

[9] Shah JJ. Parametric and feature-based CAD/CAM; concepts, techniques, and applications. John Wiley & Sons. 1995.

[10] Kosamadoudi et. al. (2013). Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements.

[11] Makris S., Mourtzis D., Chryssolouris G., Computer-Aided Manufacturing. CIRP Encyclopedia of Production Engineering.

[12] Alghazzawi (2016).

[13] Nam T.J., Wright D. The development and evaluation of Syco3D: a real-time collaborative 3D CAD System (2001).

[14] Rahiman. (2011). Impacts of VR 3D sketching on novice designers' spatial cognition in collaborative conceptual architectural design.

[15] Spatial Sketch Willis et al. (2010).

[16] DreamSketch Kazi et. al (2017).

[17] ILoveSketch Bae, Balakrishnan, Singh (2008).

[18] Settapat, Ohkura and Achalakul (2009).

[19] Wyeld, (2005).

[20] Sakatoku et al. (2009).

[21] Ni (2015).

[22] Yan & Mehta. (2018)

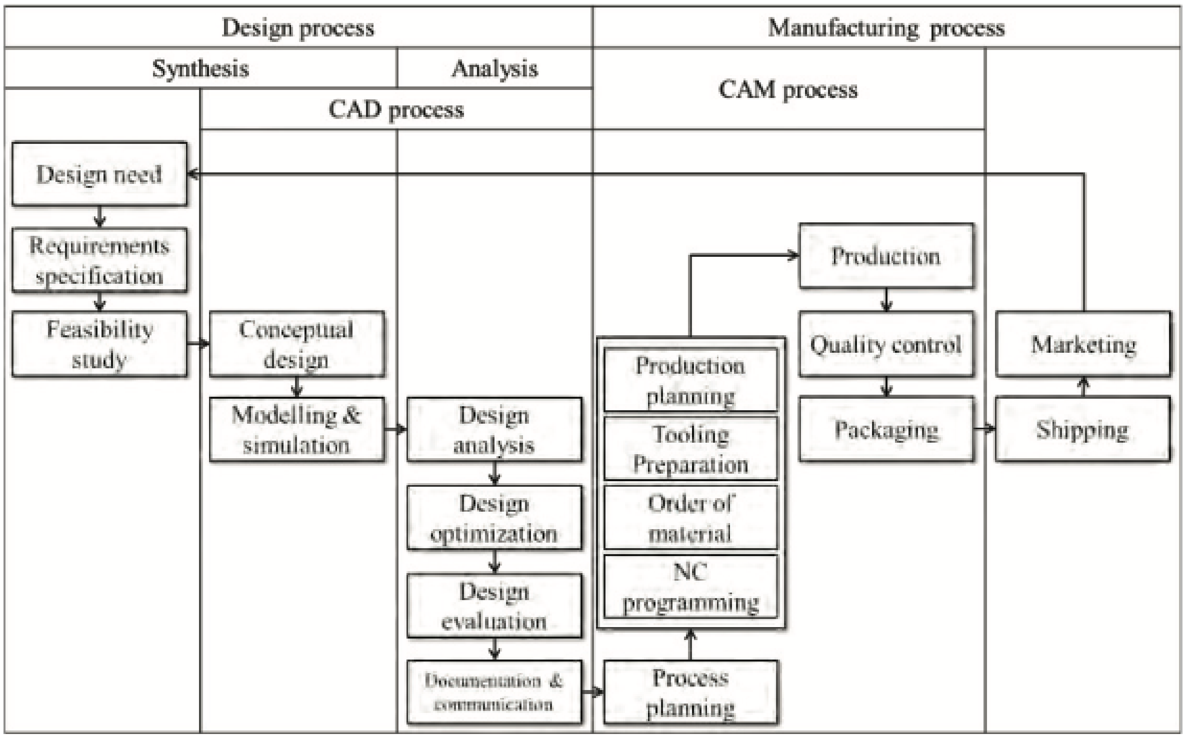
[23] Otmane', Nassima Ouramdane-Djerahl, Malik Malleml (2007).

[24] FRONT Sketch Furniture. 2006, http://www.frontdesign.se/sketch-furniture-performance-design-project
[25]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4538867
[26] https://www.youtube.com/watch?v=D5OwqXmrHS8 (in relation to https://dl.acm.org/citation.cfm?id=3224112)
[27] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1383053
[28] http://ceit.es/images/stories/files/pdf/Mechanics/paper%20hugo%20alvarez.pdf
[29] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1544670
[30] https://dl.acm.org/citation.cfm?id=503114
[31] Ferrer et al. (2013). How Does Usability Impact Motivation in Augmented Reality Serious Games for Education?. IEEE
[32] MAllem (2010). Augmented Reality: Issues, Trends and Challenges. IEEE
[33]

-Original Outline [Ignore]

Introduction:

- Preface: Advancements in Computer Aided Design has advanced user capabilities in designing and manufacturing products. It remains a core goal for CAD software to simplify a designer's ability to collect, communicate, and share creative, executable innovations, while simultaneously also minimizing human errors through mathematical optimization and precision.
    - "CADCAM is supposed to simplify engineers' task in collecting, using, creating and sharing information, but interface designed without consideration of usability often results in unsatisfied experiences and limited outcomes. Such issues can lead to low productivity, time-consuming training and deployment processes, unsatisfied designers, and safety risks."
    - Lead-in: Indirectly, with the introduction of various user-friendly innovations including wearable devices, gesture detection, and virtual reality, engineers have opened opportunities for improved HCI in CAD software. However, such innovations have been underappreciated across modern CAD technology, which has limited users' design capabilities to a 2 dimensional graphic user interface, a mouse, and a keyboard.
    - The advancement in interactive technology and relevant software enables further development in HCI across existing CAD software to enable and bridge the learning curve for novice casual designers interested in CAD. [7]
    - Current CAD software functions as a manufacturing tool for synthesizing design needs and specifications into conceptual innovations and models, and then analyzing the solution for factors such as product communication, optimization, and documentation. [7, 10, 11, 12]



From [7]

- Problem: Enhancing usability of CAD software for novice end-users. Current CAD software interfaces remain inaccessible for casual end-users, and require excessive knowledge and training that detracts from an intuitive design process easily produced through hand-drawn sketching.
    - A common obstacle users face transitioning from hand-drawn design fabrication into CAD design fabrication is the steep learning curve required to actualize a design
        - While CAD Software offers greater precision and imaging techniques, which can directly translate into manufacturing product specifications, end users are required to keep track of geometric parameters and properties in their design, including scale, dimensions, orientation, weight.
            - The excessive information detracts from the design itself. Details of such dimensions are required before a design can be created, which is counterintuitive to the design process where such properties are often finalized after a prototype is made. It has been noted in [7] that the complexity behind certain tools and libraries available in CAD imposes challenges on end-users when they "visualize, understand, and manipulate the functions through designed interaction patterns."

        - Existing user studies from [1, 2, 5] suggest that the prerequisite knowledge and emphasis on manufacturing parameters take away from the user's intention of 'designing' a product.
        - In contrast to other softwares, as CAD technology attracts a small niche community, all of whom require extensive knowledge in design, manufacturing, and geometric manipulation, little has been done thus far to resolve this persisting issue. [7]
- Traditional Graphic User Interfaces (GUI) fail to accurately model and recreate the 3D setting in which the design is to be implemented.
    - End-users communicating their designs to manufacturers, design teams, and third parties present their 3 dimensional specifications using a 2 dimensional interface. Thus, designs are separated into 2 dimensional faces of the physical object, and is susceptible to ambiguity and misinterpretation from the viewer.
    - With needs to fabricate a design as a physical object, CAD softwares, which typically operate in a virtual environment, require users to guess and estimate the practicality and feasibility of their design, as they are unable to directly construct designs at the setting in which their product will be used. [5]
    - Users who wish to use an existing product in 3D space must reproduce it on CAD before modifying and editing it. Scanning and replicating existing models are expensive, often requiring additional technology, and are often challenging to redesign and modify once exported into a CAD application. [5]
    - Users not only have to distill and synthesize informatory work pertaining to the design problem, but are also required to translate that information into a creative solution developed and communicated through the CAD UI. The level of knowledge and domain-specific expertise required to manage both processes has detracted end-users from pursuing CAD technology. [7, 8]
    - To resolve this problem, the added navigation tools such as panning, rotating, and magnification of the viewport further decreases productivity by similar reasons mentioned in 1st reason.
- The majority of existing CAD applications are not well implemented across portable and mobile devices. This limits end-users to only design on a desktop/PC environment with limited access to additional devices and hardware.
    - The majority of current software are only available on a desktop or a portable computer. Mobile devices and smaller hardware are unable to provide a streamlined user experience due to their relatively smaller screens and dimensions. While responsive designs can tailor towards individual devices, due to the immense libraries and toolkits required to manipulate objects and designs in a virtual CAD environment, such interfaces offer poorer experiences than usual CAD software technology.

        - Emphasis on Desktop and PC-based devices limit the portability of CAD technology and prevents users from developing their design remotely (outside of their typical desktop environment).
    - In contrast to typical design environments, where users typically cooperate with peers and produce design fabrications in a team, CAD software fail to offer wireless design features for users to collaborate simultaneously
        - Popularized CAD software offers ability to save and store files in a cloud-based system (e.g. Autodesk), but users are required to wait until a product has been saved before it can be modified. This is considered a bottleneck in the design development, where users are waiting until a common file is updated before reviewing the modifications and offering feedback.
        - Recent user studies have also suggested that 3D collaborative environments in CAD offer better user experiences as designers work in a team and the design process can be partitioned across each designer. [7, 13]
    - Persistence of point-click-and-drag mouse detection remained unfamiliar and non-intuitive to designers who utilize different gestures and physical interactions to fabricate designs.
        - CAD are typically devoid of gesture detection and sketch-based implementation. Thus, users with prowess in sketching struggle to apply and integrate their skills into CAD. Users who are typically proficient with hand-drawn prototypes often have to relearn skills due to the implementations and available tools in CAD software. [6]
        - Applications of gestural movements have been popularized across mobile devices (e.g. IoT), consoles (e.g. Wii), and other hardware (e.g. VR), but have yet to be utilized in CAD. To devoid users from the overemphasis on the 2 dimensional mathematical geometries and parameters, users can utilize familiar learned physical interactions and gestures to develop designs, which can reduce the steep learning curve.
        - "designers have to have a medium 'which enables half formed ideas to be expressed and to be reflected upon: to be considered, revised, developed, rejected and returned to'."

- Significance of Problem: The combination of the aforementioned problems has undermined user experiences and productivity with CAD. [7, 10]. To enhance the accessibility of CAD software, applications must offer consistent and simple methods of designing devoid from mathematical specifications and parameterization. Casual users who wish to learn and use CAD applications should be able to translate existing physical skills and experiences with hand-drawn design fabrication into CAD. Additionally, the need for clarity in design communication also means CAD applications must offer enhanced interfaces to accurately convey design fabrications in a 3 dimensional environment to manufacturers and engineers. As most design projects are now developed in design teams, CAD software should also offer utilities for users to design a product in tandem.

  - With access to augmented reality, VR, gesture detection and 3 dimensional visualization, advancements in CAD software user experience has remained stagnant. CAD Applications should offer environments for end-users to utilize elementary design knowledge and real experiences too visualize and generate 3 dimensional design fabrications in collaboration with peers, unbounded by the traditional 2 dimensional GUI.

- General Approach of Solving Problem: Using open source software and libraries that detect gesture motions, in conjunction with 3D environments utilizing VR/AR, to implement a 3D design and integration method for designing, modeling and parameterizing a design fabrication into a physical object. Specifically, identify ways to integrate technology into mobile devices.
  - Later evaluate user experience in contrast with conventional CAD software applications
  - Utilize pre-existing back-end design specification process defined using Python

- Approach Solving Solution: Existing gesture detection libraries offer ability to transform gestures into detectable physics and mathematical vectors that translate into sketches using Virtual Reality. Additionally, nature of AR/VR offers ability to directly view product in 3D space. Finally, use of mobile devices and/or VR offer ability to communicate wirelessly with other device holders, enabling potential opportunities for wireless interactions in one single 3D design environment.

- Value of Approach: Developing gesture detection for CAD can be extended beyond product design and translate into other forms of CAD such as graphic design. In addition, relevancy of gesture detection-based design and editing can be extended beyond design into other editing software and medium such as video editing and game design. Successful understanding and integration of gesture detection in software editing can devoid users from the simply mouse and keyboard system, to use familiar physical experiences as a guide for enhanced user experience.

Compelling Reasons for Project: Problem Specification

1. Learning Curve and Accessibility of CAD Software for Novice Designers
   Summary: Popular CAD Applications offer pre-defined tools and utilities that require understanding of mathematical concepts and software-specific utilities. Such implementations increases the learning curve for users, and forces users seek non-intuitive ways to actualize their designs.

Scenario: Learning how to use a CAD Software; translating hand-drawn designs into a CAD-based design; applying knowledge from one CAD software into other CAD softwares; usability of CAD software as a casual end-user.

- **Overemphasis on Design-Based Parameters:** The emphasis on CAD-specific parameters such as coordinates and geometry, takes away from the user's intention of 'designing' a product, instead transforming the design directly into a product with intended specifications. The emphasis of specifying parameters in CAD can take away from the design process, which focuses on providing a generic sketch of an intended design solution to the problem at hand. The math dictates the design, instead of the other way around. Notable challenge in producing an easy-to-use designing package that takes little training. [1, 2, 5]
- **Library Limitations:** While pre-programmed libraries, databases and templates for simple geometries are available in most CAD softwares, for complex geometries that are more effectively communicated through a sketch or hand-drawn design, users have to "mix-and-match" what is available in the CAD toolkit, which complicates the design process. Thus, a casual user's creative freedom is limited to the predefined resources available.

2. GUI Implementation of CAD in 3D Space for Design Fabrication

   Summary: Fabricating designs in a virtual environment removes users from the physical implementation of their product in 3D Space. To ensure the product can be pragmatically manufactured, CAD Software should offer inexpensive utilities to directly implement and design products in a 3D environment, where casual users can utilize pre-existing models as a guide.

   Scenario: Developing a physical object in 3D space using CAD Softwares, which typically offer only GUI; implementing design specifications into CAD software; redesigning an existing 3D model of a product (or using the existing product as a guide for future designs).

   - **Limitations of Traditional GUI:** Increasing trends in designing 3D physical objects through CAD software suggest that design fabrication is currently undervalued as an 'output method from existing CAD applications, without specific consideration given to interface design' [4, 5]
   - **Translating 2D into 3D:** With needs to fabricate a design as a physical object, CAD softwares, which typically operate in a virtual environment, require users to guess and estimate the practicality and feasibility of their design, unable to directly construct designs at the setting in which their product will be used. [5]
   - **Expensive Replication:** Users who wish to use an existing product in 3D space must reproduce it on CAD before modifying and editing it. This process takes away from the actual design and development process, which is a limitation of existing CAD software. Scanning and replicating existing models are expensive, often requiring additional technology, and are often challenging to redesign and modify once exported into a CAD application. [5]

3. Portability of CAD Software across Devices

   Summary: The majority of existing CAD applications are not well implemented across portable and mobile devices. This limits end-users to only design on a desktop/PC environment with limited access to additional devices and hardware.

Scenario: Fabricate and develop designs on a tablet, phone, VR (anything that isn't desktop or laptop-based); designing remotely outside of the desktop environment, using a portable device

- **Desktop Environment:** Emphasis on a "Desktop/Computer Environment" with little use of Tablets, Mobile Devices, sensors, etc. that are more intuitive and trending in contemporary technology. This makes CAD software and digital design fabrication less portable compared to other softwares.
- **Responsive UI Challenges:** With increased use of alternative devices outside of laptops and desktops, it is imperative that CAD software further tailors its UI to accommodate for the different capabilities of tablets, mobile devices, and even hardware such as VR devices. Currently, CAD softwares are Desktop specific, because other user interfaces do not offer sufficient space for designs to be updated or modified. Additionally, responsive UI are impractical for CAD software, due to the large number of tools required for a typical CAD software, and the limited space available in a mobile device screen to display and interact with such tools. This limits opportunities for designs to be portable and for users to design remotely on other devices.
- **Design Across Multiple CAD Applications:** Migrating between CAD software or using multiple CAD software for design is impractical as different softwares have different GUI implementations that utilize different design libraries and logic. Thus, design fabrication using CAD applications are not portable. There is inconsistency between different CAD applications. (e.g. the idea of splining in Autodesk 360 is native to the Autodesk series and is unavailable in SketchUp). For users looking to migrate between platforms, this inconsistency imposes a challenge on users and lacks accessibility for new individuals looking to integrate CAD into their design work.

4. Lack of Cooperative Utility/Multi-user Interaction in CAD Software

Summary: Existing CAD Applications have yet to explore multi-user interfaces and wireless capabilities for design fabrication. These applications limit users from accurately communicating their designs, which is imperative for manufacturing and engineering the product.

Scenario: Fabricating a design in a team; presenting designs in a 3D environment and communicating designs to manufacturer/design team

- **Cooperation:** Typically, CAD software are user-specific, where design files are exported and sent back and forth between users to communicate, update, and modify designs. The lack of wireless capabilities for multiple users to fabricate a design in parallel means that design teams have to wait until a CAD file is saved before modifying or offering feedback. Alternatively, multiple designers can cooperate and edit a design concurrently by sharing the same desktop, which is counterintuitive.
- **Design Communication:** Knowledge and data pertaining to the design has to be clearly communicated across engineers, manufacturers, and design teams, which typically exist in isolation from each other. "Thus graphics vocabulary used in the design process is dramatically different from the one used when the product is being manufactured and assembled., and has to be presented in a clear, accessible manner." [5]. The 2 dimensional perspective traditional GUI means that manufacturers and co-designers have to imagine how the design will be implemented in the actual design setting, which is susceptible to ambiguity.

5. Limited Integration between Gesture Detection and CAD Software

Summary: Gesture detection can simplify complicated CAD-based design implementations, which often use point-click-and-drag interfaces. By integrating embodied physical interactions into CAD applications, skills specific to hand-drawn designs, such as sketching, can be easily utilized into CAD software, making the technology more intuitive, natural, and accessible for casual end-users.

Scenario: Applying sketching skills into CAD; transforming design ideas into a sketch; integrating properties of gesture detection into CAD; Designing complex geometries with CAD in 3D space vs Sketching it by hand

- **Benefits of Sketching:** CAD are typically devoid of gesture detection and sketch-based implementation. Thus, users with prowess in sketching struggle to apply and integrate their skills into CAD. Users who are typically proficient with hand-drawn prototypes often have to relearn skills due to the implementations and available tools in CAD software. [6]
- **Existing Interfaces:** Gestural movements are trending and available across mobile devices (e.g. IoT), but have yet to be utilized in CAD. On a user experience level, the point and click system can be tedious when it could easily be replaced with a gesture detection.