UNIVERSITY OF KENTUCKY

College of Engineering, Department of Computer Science

Project Pitch Cover Page

Title of the project: Building a TTD GAI for Automation in Computer-Aided Design (3D) and Engineering Drawing (2.5D).

Shortened Name of Project: TTD GAI

Principal Investigator:
Dr. Fuhua (Frank) Cheng
Department of Computer Science, College of Engineering
Email: cheng@csuky.edu
Phone No: (859) 948-3375

Total Amount Requested: about $1.8m for 18 months. Details will be given in the full proposal

Contents:

Summary of the Project: The purpose of this project is to develop a 'Text-to-Design Generative Artificial Intelligence (TTD GAI) system called OPUNA that can be operated using natural language prompts for automation in computer aided design (3D) and engineering drawing (2.5) or even various everyday life events.

Acronyms: TTD (text-to-design), GAI (generative artificial Intelligence)

Signature:

Dr. Fuhua (Frank) Cheng
Principal Investigator

1. Technology Innovation

(Explain the core high-risk technical innovation to be researched and developed during a Phase I project. NSF must understand what research and development is required and how this technical innovation differs from and is significantly better than existing solutions. It may also be that the proposed innovation creates a new market. In this case, why will it be adopted? Describing features or benefits of the proposed technology is not sufficient.)

AI has evolved from traditional AI to Generative AI (GAI) and, recently, to AI Agents.

However, they all have a common problem: they focus more on cognition-related tasks, less on perception-related tasks. One reason is, it is easier to handle data and tools in the first case than the second case. Consider, for instance, the structure of an AI Agent (see Fig. 1).

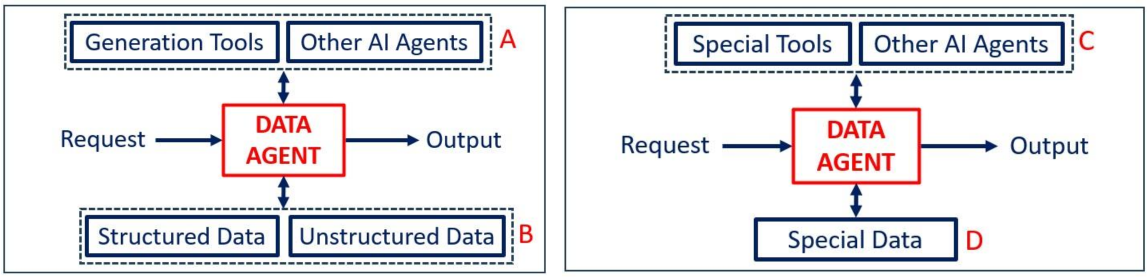

Fig. 1

Cognitive tasks involve processes like learning, reasoning, problem-solving, planning, decision-making, and understanding language. To ensure these processes are performed properly, the system builders put as much information into the data box and the tool box (see A and B in the left figure) as possible. This leads to huge data set and tool set, so huge that are completely beyond our imagination. That is why they have to use millions of GPUs to run the data centers. But, most of the time, most of the resources (time and electricity) used in doing the learning, reasoning and planning processes are wasted because not every task needs to go through the entire data set and tool set to finish those processes.

On the other hand, perception tasks involve gathering, interpreting, and processing sensory information from the user or the environment. This is a much smaller scope but needs to prepare special data set and tool set (see C and D in the right figure) for the data agent to function properly. Preparing the special data set and tool set is certainly more difficult than simply putting all the data items and tools you can get into the data set and the tool set. That is why most of the existing AI systems can only handle cognitive tasks and can only provide text-based content on patterns in existing data. Actually, there hasn't been any powerful AI systems developed solely for perceptive tasks yet. The above right figure is our view of the structure of such an AI Agent.

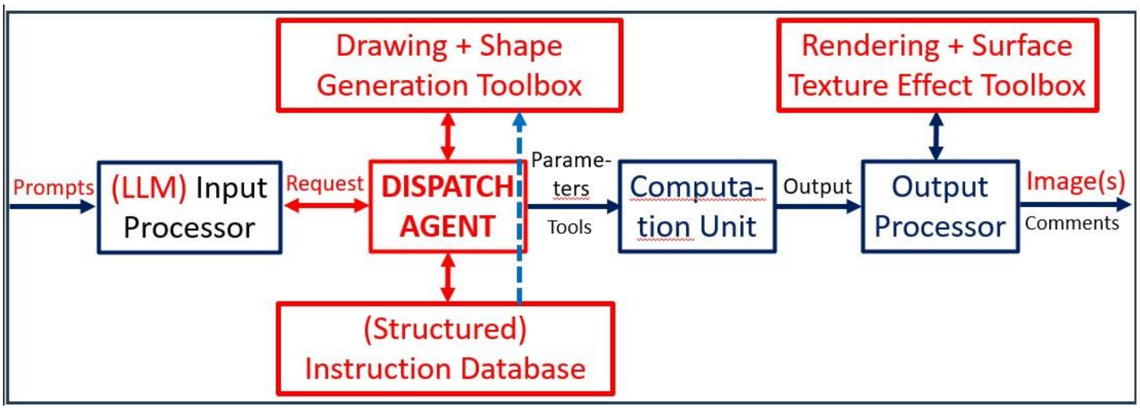

Fig. 2

As far as the capability of providing image-based content as output is concerned, SORA of openAI is the only significant player in this category; SORA is a TTV (text-to-video) GAI that can generate video of one minute long based on the provided prompts. The scenario of the generated video is quite impressive. But the video cannot be edited and one cannot judge the precision of the prompt-to-video conversion.

Yet, there are many occasions where we need AI systems that not only can handle special type perception tasks of the event, but can also generate image-based content as output for the event.

In this phase I project we will build a **TTD** (text-to-design) **AI agent** (**GA**I) for automation in computer aided design (3D) and engineering drawing (2.5D). The drawing part can be used for ordinary life events as well. The AI agent will be built based on a new concept shown in Fig. 2. The core of this new concept is the part called Dispatch Agent. Different from an ordinary AI agent which does **sensing**, r**easoning**, **planning**, c**oordination**, and **actin**g all by itself, the Dispatch Agent will do three things only: ***reasonin***g, ***plannin***g and ***coordinating***. The sensing will be done by a separate **LLM** in the **Input Processor**, the acting will be done by a separate Computation Unit and a separate Output Processor. The Input Processor, the computation unit and the Output Processor will do their work on GPUs. There are two advantages of this arrangement. First, since the LLM is not part of the Dispatch Agent, no learning/training will be performed on the data set (in this case, an Instruction Database) and the tool set. The second advantage is that prompts that have been processed by the LLM can be sent to the Dispatch Agent immediately (in the form of requests) so the Dispatch Agent can do the reasoning, planning and coordinating work for these requests while the LLM processes the remaining prompts in the queue. Likewise, requests whose construction plans and implementation plans have been built by the Dispatch Agent can be sent for computation and rendering immediately so the Dispatch Agent can build construction plans and implementation plans for new requests sent by the Input Processor.

Note that in Fig. 2, there is no such a thing as a **Data Set,** but an **Instruction Database**. The Instruction Database stores all the commands needed to develop a construction plan for generating a 2D or 3D scene and the **Drawing + Shape Generation Toolbox** stores all the functions and tools to have the construction plan implemented. Each command in the Instruction Database has pointers to the functions and tools in the Drawing + Shape Generation Toolbox that are needed for the implementation of that command (see the dotted blue arrow that goes from the Instruction Database to the Toolbox). So, the coordination job of the Dispatch Agent is straightforward.

The Dispatch Agent does reasoning and planning partially through the LLM in the Input Processor (see the two-way arrow between the input processor and the Dispatch Agent) but mainly through hashing using a two-way communication channel with the Instruction Database. If one of the prompts says: generate a 3D object by joining two cylinders with these (…) orientations and these (…) dimensions (see the portion inside the dotted box in the left figure in Fig. 3), the Dispatch Agent will get a request from the LLM for it to identify an appropriate command in the Instruction Database to perform a Boolean union operation on two cylinders with the specified orientations and dimensions, and then through the pointers in the command to identify all the transformation functions needed to transform the standard cylinders to the specified locations with specified dimensions and build a corresponding Extended Constructive Solid Geometry (ECSG) tree with the transformation information and the required Boolean union operation, then the computation unit and the output processor can do the ray tracing job and the rendering job so the image of the object E as the one shown in the dotted box in Fig. 3 can be generated.

What the above process shows is: all the things an experienced CAD engineer must do with a CAD software package such as AutoCAD to generate the object E shown in the dotted box of Fig. 3 can now be done with just one simple prompt.

2. Technical Objectives and Challenges

(Clearly explain the specific research and development required to prove that the foundational technology works and address the associated challenges explicitly with a high level description of how each will be managed. This section must convey how the proposed work is technically innovative and demonstrate that you have an understanding of the core research and development tasks necessary to prove out the technical innovation.)

Five issues have to be addressed here: how should each 2D or 3D scene be constructed, how should each 2D or 3D scene be represented, how should the reasoning, planning and coordinating job of the Dispatch Agent be performed, how should a 2D or 3D scene be rendered, and how should the *instruction database* and the *drawing + shape generation* (DSG) *toolbox* be built.

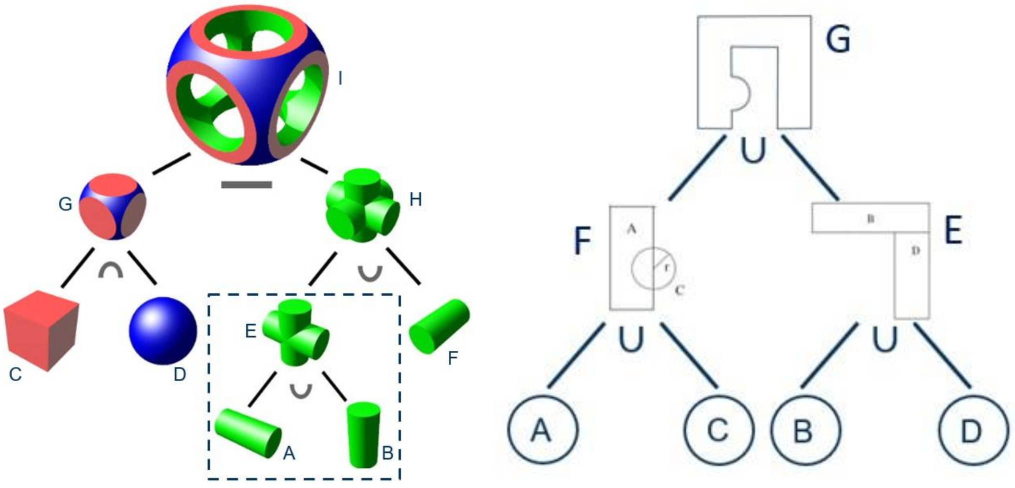

Fig. 3

Two types of construction mode will be supported: Boolean Mode (BM) and Freeform Mode (FFM). A BM performs a regularized Boolean operation (union, intersection or difference) on two primitive components or results of other construction steps. A FFM performs freeform shape construction using curve interpolation, surface interpolation or surface blending (see the dark blue surfaces in Fig. 4 for blending examples). Each 2D or 3D scene will be constructed through an ECSG tree building process level by level. Each construction step is to perform a BM operation or a FFM operation. See Fig. 3 for the construction of a 3D object and a 2D object using just BM operations. The Dispatch Agent will keep track of all the construction steps until the process is halted by the user. The result of each construction step will be assigned a temporary ID (see both figures in Fig. 3) so that when the result of a construction step is used as a component of a new construction step, the Dispatch Agent knows where to find that component. Fig. 5 shows examples of BM operations performed on results of some FFM operations.

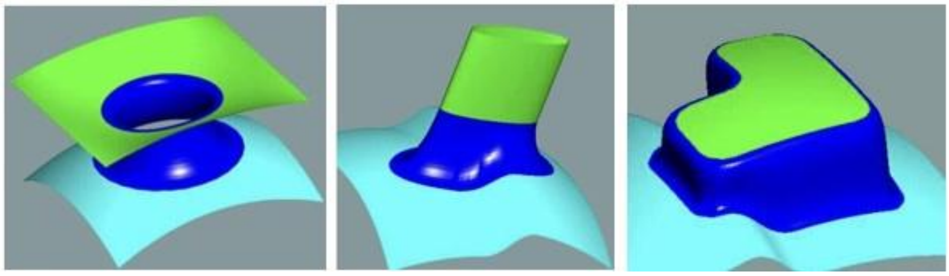

Fig. 4

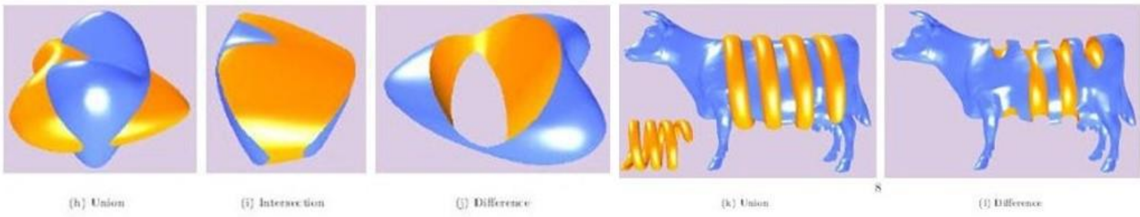

Fig. 5

Each 2D or 3D scene will be represented as a set of ECSG trees. A ECSG tree is a CSG tree whose internal nodes are not just BM nodes but can also be FFM nodes. This representation scheme seems to be the best choice because the representation not only shows a way to decompose a scene or an object but also reflects the construction process of the scene or the object. To make the subsequent rendering process more efficient, each node of the ECSG tree will also contain a bounding box of the corresponding component of that node. This representation scheme is not unique though.

The Instruction Database stores all the commands needed to perform all the BM and FFM operations and drawing operations for 2D or 3D scenes. Each command contains operation category and type, required parameters, required transformations and tools to perform the operation, and pointers to those transformations and tools in the DSG toolbox. Therefore, there is no need for the Dispatch Agent to do a search of the DSG toolbox to find the required transformations and tools for the implementation of a command and, consequently, the DSG toolbox can be arranged in anyway that would fit our work need. A BM command does not need any tools in the DSG toolbox except transformation functions to transform the components to the specified location with the specified orientation and dimension, only a FFM command will need to use some tools in the *DSG toolbox* to build an interpolating curve, an interpolating surface or a blending surface. The structure of each command is shown in Fig. 6. Each of the green, black, red and blue boxes is of fixed length. Therefore, the Dispatch Agent can simply *hash* into the instruction Database to get the needed command once it has been reasoned out what operation is needed for the input prompt.

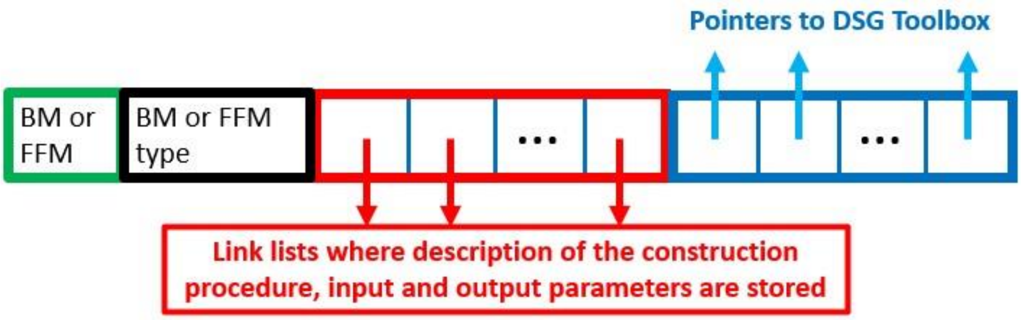

Fig. 6

With the above arrangement of the commands in the Instruction Database and the connection of each command with the DSG toolbox, the planning and the coordinating job of the Dispatch Agent is straightforward, actually of complexity O(1) in each case. The reasoning job will mainly be done by an LLM which is separate from the Dispatch Agent. There are several free and open source LLMs that we can use for our work here. Popular cases include: Llama series (Meta), Qwen series (Alibaba), Gemma (Google DeepMind) and Phi series (Microsoft). These models are all available to download on platforms like Hugging Face. Llama 2 and Llama 3 are good choices for our work because they are ready for research and commercial use. The reasoning job here is not as complicated anyway because the prompts received by the LLM all have the same goal: to build an geometric object or a component of a geometric object. The main subject is just to figure out what geometric operation the prompt expects the system to perform. Another advantage here is: editing is easy (see below) so any misinterpretation can be fixed immediately.

The Computation Unit and the Output Processor are separate from the Dispatch Agent (see Fig. 2). The Computation Unit receives construction and implementation plans for each node of a ECSG tree level by level. The ECSG tree represents the construction process of a 2D or 3D scene. If the node is a BM node, the Computation Unit will perform the required transformations on the components of the BM node. If the node is a FFM node, depending on the type of the FFM node, the Computation Unit will generate a Beta-Bezier curve or surface to interpolate a set of given data points with given tension parameters, or generate a Beta-Bezier surface to smoothly connect two trimmed NURBS surfaces at the trimming curves with tension control (called a blending process, see Fig. 4). The result of the Computation Unit is sent to the Output Processor to perform a Z-Buffer process for the result of a FFM node or a ray tracing process for the result of a BM node. The rendering process of the Output Processor follows the flow of OpenGL. The Output Process as well as the Computation Unit and the Input Processor all do their work on GPUs. Three choices are available to program the GPUs: Compute Shader, CUDA and OpenCL. The Compute Shader coding language will be used here since it can be used for all the GPUs available in the market, not just Nvidia’s GPUs.

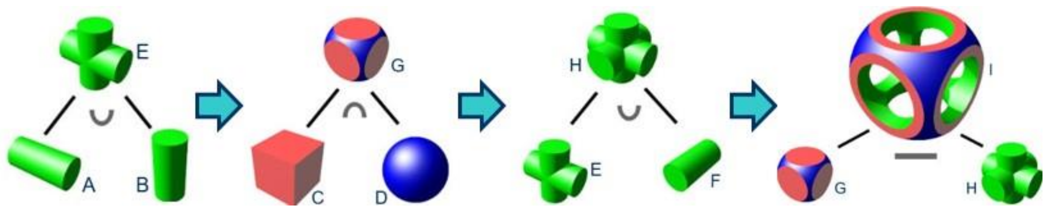

Fig. 7

The design shown in Fig. 2 provides a rendering procedure that is easy to do editing. Note that the system creates a 2D or 3D scene through a process that is like building a tree in a bottom-up and left-to-right fashion and result of each step in the building process is shown on screen, so it is convenient for a user to check if the prompts are accurately interpreted and executed. The case shown in the left figure of Fig. 3 can be used as an example to illustrate why this is so. The construction of the object start with building component E. Once this is done, result of this step is rendered and shown on screen as the left most item in Fig. 7. The process then continues with building component G, and then component H and finally the object I. Results of these steps are also rendered and shown on screen in that order (See Fig. 7). Therefore the user can check immediately if each step is indeed interpreted and executed accurately. Editing can be implemented simply by providing an undo operation for the system to repeat the step again or asking the user to provide a more suitable prompt.

So all the issues have been successfully addressed.

3. Market Opportunity

(Explain the value of the technological innovation including the potential uses and those who will benefit (who is the customer) and demonstrate a high-level understanding of the competitive landscape and why this innovation has the potential to compete.)

According to our investigation, the following three patents are related to CSG tree representation:

1. US5760786A: “Simultaneous constructive solid geometry (CSG) modeling for multiple objects" (expired on 2015/12/27).
2. US6392645B1: “Three dimensional geometric modeling system" (expired on 2019/03/15).
3. US5537519A: “System and Method for Converting Boundary Representations to Constructive Solid Geometry Representations for Three-Dimensional Solid Object Modeling” (expired on 2013/07/16).

The new representation technique “Extended Constructive Solid Geometry (ECSG) Tree” proposed in this project pitch does not infringe these patents, not only because these patents have all expired, but also because none of these patents include FFM nodes in the CSG tree.

On the other hand, all the shape generation techniques such as Beta-Bezier curve interpolation, Beta-Bezier surface interpolation and the surface blending techniques used in this project are published research results of the project team members. Therefore, the representation and computation techniques of this project do not have any intellectual property problem.

As far as competitive landscape is concerned, other than SORA which is a text-to-video GAI mainly for casual use purpose, existing prominent text-to-image GAIs on the market include:

DALL-E 3: known for generating high-quality images from text descriptions.
Midjourney: also known for creating visually impressive images.
Runway: can do both text-to-image and text-to-video.
Adobe Firefly: can do text-to-image and other generative AI features.
Stable Diffusion: popular open-source text-to-image model.

Again, none of these models are for design purpose, they are, like SORA, mainly for casual use. Hence, at least on the surface, the text-to-design area seems to be a field without too many players.

The TTD GAI to be developed in this project can be used for both *ordinary life events* such as drawing a 2D or 3D map for a natural disaster in a rural area and *engineering applications* such as mechanical CAD or circuit CAD. Possible customers include both ***individuals*** and ***groups.*** For engineering applications, individual users could include both professional CAD engineers and hobbyist users. The reason that a professional CAD engineer would use a

TTD GAI to do his/her work is the same as an experienced car driver would use an unmanned vehicle for his/her communication need. The only criterion is if the GAI has a trustworthy performance.

The customers of a TTD GAI could include AI models. For an AI model to step into the mechanical CAD or the circuit CAD area, it must have the capability to generate image-based content. Actually such a capability is needed in many other areas as well, such as art or surgical, to name just a few. So a good AI model would definitely be interested in integrating a good TTD GAI into its system. Prominent mechanical CAD and circuit CAD companies such as AutoCAD, SolidWorks, Autodesk, Cadence and Siemens certainly wouldn't want this to happen because that would mean less business for their CAD packages. But we believe this is the inevitability of the trend.

With all these information, we think the potential of this TTD GAI's market opportunity is quite good.

4. Company and Team

(Explain the team's suitability to successfully execute the project based on the proposed innovation and approach to R&D. Provide information on plans to address gaps in the team.)

Company

Name:
Location:
IRS EIN No:

Current Team

Fuhua (Frank) Cheng (US citizen) : PhD (Ohio State University), Research areas: computer graphics, geometric modeling
Brent Harrison (US citizen): PhD (North Carolina State University), Research areas: large language models, generative AI, human centered AI, safe|ethical AI
Anastasia Kazadi (US citizen) : PhD (University of Kentucky), Research areas: computer graphics
Shuhua Lai (US permanent resident) : PhD (University of Kentucky), Research areas: computer graphics, geometric modeling
Alice Lin (US citizen) : PhD (University of Kentucky), Research areas: computer graphics
Kenji Miura (citizen of Japan) : PhD (Cornell University), Research areas: computer graphics, geometric modeling
Seif Moustafa (US citizen) : MS (University of Kentucky), Research areas: computer graphics, geometric modeling
Wyatt Scott (US citizen) : MS (University of Kentucky), Research areas: computer graphics
MdFayek Sharaf (US permanent resident) : BS (University of Kentucky), Research areas: computer graphics

Team's suitability to successfully execute the project

A focused core team in artificial intelligence, computational geometry, GPU programming, rendering, and system integration will execute early prototyping. A person (xxxx xxxx) with natural language processing ability will lead the development of the Input Processor, ensuring that natural language prompts are reliably transformed into structured requests for design generation. A computational geometry expert (xxxx xxxx) will design and implement the algorithms that support Extended Constructive Solid Geometry, Boolean operations, and advanced curve and surface generation, providing the mathematical foundation for construction planning.

To achieve the necessary computational efficiency, the person with GPU programming ability (xxxx xxxx) will develop high-performance implementations of the Computation Unit and Output Processor, utilizing Compute Shaders to ensure broad compatibility across hardware platforms. A person with rendering experience (xxxx xxxx) will be responsible for visualization and implementing ray tracing, Z-buffering, and OpenGL-based rendering pipelines to produce accurate and responsive outputs. The person overseeing the integration of these efforts (xxxx xxxx) will design the overall framework, ensuring seamless interaction among the Input Processor, Dispatch Agent, Computation Unit, and Output Processor.

Our streamlined team provides the critical capabilities needed to demonstrate proof of concept for the Text-to-Design Generative AI system, laying the groundwork for subsequent development and expansion.