Eurographics Conference on Visualization (EuroVis) 2023
D. Archambault, R. Bujack, and T. Schreck
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 3

VENUS: A Geometrical Representation for Quantum State Visualization

Shaolun Ruan[1], Ribo Yuan[2,1], Qiang Guan[3], Yanna Lin[4,1], Ying Mao[5], Weiwen Jiang[6], Zhepeng Wang[6], Wei Xu[7], and Yong Wang[1†]

[1]School of Computing and Information System, Singapore Management University, Singapore
[2]Department of Computer and Information Sciences, University of Delaware, United States
[3]Department of Computer Science, Kent State University, United States
[4]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China
[5]Computer and Information Science Department, Fordham University, United States
[6]Electrical and Computer Engineering, George Mason University, United States
[7]Computational Science Initiative, Brookhaven National Laboratory, United States

Abstract

Visualizations have played a crucial role in helping quantum computing users explore quantum states in various quantum computing applications. Among them, Bloch Sphere is the widely-used visualization for showing quantum states, which leverages angles to represent quantum amplitudes. However, it cannot support the visualization of quantum entanglement and superposition, the two essential properties of quantum computing. To address this issue, we propose VENUS, a novel visualization for quantum state representation. By explicitly correlating 2D geometric shapes based on the math foundation of quantum computing characteristics, VENUS effectively represents quantum amplitudes of both the single qubit and two qubits for quantum entanglement. Also, we use multiple coordinated semicircles to naturally encode probability distribution, making the quantum superposition intuitive to analyze. We conducted two well-designed case studies and an in-depth expert interview to evaluate the usefulness and effectiveness of VENUS. The result shows that VENUS can effectively facilitate the exploration of quantum states for the single qubit and two qubits.

CCS Concepts

• ***Human-centered computing*** → *Visualization application domains;* • ***Hardware*** → *Quantum computation;*

1. Introduction

Quantum computing has been undergoing impressive development in recent years [AAB*19, MV17]. An increasing number of applications have been proven possible to achieve quantum speedups such as optimization problems, machine learning, cryptography, and finance [HCS*20]. Meanwhile, along with the evolution of real quantum computers provided by many quantum vendors such as IBM, Rigetti, Honeywell, and IonQ [SAP*21], the last decade represents a significant milestone in the field of quantum computing [CKJA*19, TCCG20, Gom18].

For the implementation of quantum computing, a quantum bit (*a.k.a. qubit*) is the basic unit of any quantum program. Generally, the state of a single qubit or multiple qubits is called *quantum state*, such as state $|0\rangle$ for a single-qubit state and state $|01\rangle$ for a two-qubit system. Compared to classical computing, today's quantum computing is driven by two basic quantum-specific properties called *quantum superposition* and *quantum entanglement* [RP11, TQ19], which makes it possible to achieve the power of the quantum advantages [Ste98]. Specifically, quantum superposition (*a.k.a., superposition*) indicates the unknown quantum states before measurement. Upon measurement of the qubit, it collapses to either the $|0\rangle$ or $|1\rangle$ state according to the deterministic probabilities of all possible states [GJAE*20]. For example, the probability of the measured result 0 is 0.4, while that of 1 is 0.6. Also, quantum entanglement (*a.k.a. entanglement*) supports the correlations between multiple single qubits such that manipulating one qubit can impact the state of the other qubit(s). Among them, two-qubit entanglement plays a critical role in many popular applications (*e.g.*, Grover's Algorithm [Hay08], Deutsch Algorithm [GRL*03], and Quantum Machine Learning [SLM*21, BBF*20]), especially in the near-term Noisy Intermediate-Scale Quantum [BCLK*21] (NISQ) era where the qubit number is severely limited.

To effectively study and execute quantum programs, observing the quantum state is a basic requirement. A visualization technique named *Bloch Sphere* [Blo46] was proposed to meet this need. *Bloch Sphere* leverages a point on the unit sphere to represent

† Yong Wang is the corresponding author.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

the *amplitudes* of a pure single-qubit state. Due to the intuitiveness, *Bloch Sphere* is widely-accepted in the quantum computing community to visually show the quantum state of a single qubit [Wie14, FA09, HD04] and has been incorporated into many popular toolkits of quantum computing, such as Qiskit by IBM Quantum [ibm14]. Despite its prevalence, several issues still exist when using *Bloch Sphere*. First, it is impossible to use *Bloch Sphere* to visualize quantum states of more than one qubit [BSR21, Wie14]. Once two qubits are entangled, the visualization approach *Bloch Sphere* breaks down because the quantum state cannot be condensed into three dimensions in the same form [ZHVG20]. Second, although *Bloch Sphere* can visualize quantum states, it cannot support an intuitive view of the probability of all basis states. However, the probability is a critical property of quantum states as it is used to directly reflect superposition [Wie20]. Third, three-dimension visualizations, like *Bloch Sphere*, have been proven to perform worse than two-dimension counterparts when conducting precise measurements [TKAM05, ARH*15]. The prior study has found that the three-dimensional visualization of the *Bloch Sphere* introduces the possibility of visual occlusion [Wil21], which makes the observation of quantum states rather inaccurate.

Thus, a novel 2D visualization approach is urgently needed to address the above issues for all quantum computing users (*e.g.*, novices and experts). It is a challenging task to visualize quantum states and the challenges mainly come from two aspects: *individual properties of quantum states*, and *complex construction of quantum entanglement*. First, according to our co-design process with domain experts, it is essential to visually identify the correlation among all quantum state components (*e.g.*, the amplitudes of the state vectors and probabilities of quantum states) other than representing the above components individually. Because this can make users aware of the hidden logic of the evolution of quantum states. However, how to seamlessly link the individual quantum state components by appropriate visual channels is not a trivial task. Second, how to visualize the entangled quantum states remains a challenging task. Given that the two-qubit entanglement state is a specific type of two-qubit state, the two-qubit state representation is still non-transparent for quantum computing researchers and developers since the two-qubit state is not a simple accumulation of multiple single-qubit states. Meanwhile, how to utilize the same scalable form of visualization to represent the entangled states of both single qubits and multiple qubits would be even more difficult.

To address the above issues, we propose *VENUS*, a novel **V**isual d**E**sig**N** for quant**U**m state repre**S**entation. *VENUS* supports quantum state representation for not only single-qubit states but also two-qubit states, which is the basis for two-qubit entanglement representation. Meanwhile, *VENUS* can inform users of the probability for the quantum superposition without any manual calculation of the probability, leading to more efficient and smooth analysis of arbitrary quantum states. Following a user-centered design process [Mun09], we work closely with five domain experts in quantum computing. A co-design process is conducted to iteratively derive design requirements for observing quantum states, which also guide our subsequent visual design. *VENUS* mainly consists of two coordinated visual components: a set of right triangles to visualize state vectors, and the circumscribed semicircles of these right triangles explicitly reflect the probability distribution of the corresponding quantum states. To evaluate the usefulness and effectiveness of *VENUS*, we present two case studies on single-qubit (*i.e.*, Quantum Neural Network) and two-qubit (*i.e.*, *Grover's algorithm*) and an in-depth interview with 14 domain experts. The results show that *VENUS* can effectively facilitate the visual analysis of various quantum programs. The major contributions of this paper can be summarized as follows:

- We co-design with five domain experts to identify and summarize the design requirements for visually analyzing the quantum states, pinpointing the common challenges when performing visual analysis of quantum states.
- We present a novel visualization, *VENUS*, to tackle both quantum superposition and quantum entanglement representation challenges for both single-qubit and two-qubit scenarios with multiple visually correlated 2D geometrical shapes.
- We conduct two case studies and user interviews with 14 domain experts to validate the effectiveness and usefulness of *VENUS*.

2. Background

In this section, we introduce a set of relevant concepts in quantum computing, including the building block of quantum computing, the properties of the qubit, and the quantum state of the qubit.

2.1. Building Block of Quantum Computing

Qubit, the quantum version of the classic bit, is the basic unit in quantum computing. Similar to a classical bit, there are two computational basis states called state 0 and state 1 for a qubit [HIP*21]. However, a qubit can also be in an arbitrary linear superposition of state 0 and state 1 [HIP*21, RP00], which is well-known as quantum superposition. Mathematically, one can represent a qubit using the form of a state vector [RP00].

Quantum gate, just like the manipulation of classical bits using classical logic instructions such as *OR* and *AND*, it is applied to qubits to change their quantum states. Mathematically, quantum gates are represented as operation matrices, acting on single qubit or multiple qubits. Operations of quantum gates are equivalent to the dot products with the state vector of qubits.

Quantum circuit, similar to the classical circuit, is the implementation of the quantum program for execution. A quantum circuit consists of a set of quantum gates, acting on one or multiple fixed qubits. As shown in Figure 3 and 4, a quantum state will be initialized from the start of the quantum circuit and manipulated by quantum gates designed in the quantum circuit.

2.2. Properties of Qubit

Superposition indicates that a qubit can not only be in one of the computational basis states $|0\rangle$ or $|1\rangle$, but also in a linear superposition of this two states [Nar99]. Thus, the quantum state $|\psi\rangle$ of a qubit is described by $\alpha|0\rangle + \beta|1\rangle$, where the complex numbers α and β are called *amplitudes* such that $|\alpha|^2 + |\beta|^2 = 1$ [RP00]. Meanwhile, the probability of measuring $|0\rangle$ is $|\alpha|^2$ and the probability of $|1\rangle$ is $|\beta|^2$ [HIP*21, RP00, Nar99]. as a quantum system with n qubits can generate a linear superposition of 2^n basis states simultaneously [RP00, Hey99].

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Entanglement is an essential feature that differentiates qubits and classical bits. Specifically, when two or more qubits are entangled, their quantum states are coupled with each other, so that changing the quantum state of any one qubit will instantaneously change the other qubit's quantum state in a predictable way [RP00]. Note that the entanglement operation requires more than one qubit, making it critically important to analyze the quantum states of multiple qubits instead of a single qubit.

2.3. Quantum State of Qubit

In quantum computing, a quantum state is a mathematical entity that provides a probability distribution of different basis states. For clarity, we start with a single-qubit state, and the case of a two-qubit state will be derived from these results. In Section 5, we will illustrate how we apply the following quantum computing characteristics and encode them with a variety of 2D geometric shapes to form our visual design.

Single-qubit state represents the quantum state for a single qubit. Recall that the quantum state of a qubit can be a superposition of basis states $|0\rangle$ and $|1\rangle$, thus the quantum state $|\psi\rangle$ can be expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\begin{bmatrix}1\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$, where the amplitudes α and β satisfy:

$$\alpha = a + b\cdot i, \beta = c + d\cdot i, \tag{1}$$

where i is the imaginary unit, and a, b, c, and d are real numbers. Based on the quantum theory, the probability of a measured quantum state (*e.g.*, $|0\rangle$) satisfies

$$Pr(|0\rangle) = |\alpha|^2 = |a|^2 + |b|^2. \tag{2}$$

Meanwhile, since the amplitudes satisfy a normalization constraint,*i.e.*, the sum of the probabilities of the two quantum states for single qubits (*i.e.*, $|0\rangle$ and $|1\rangle$) consistently equals 1, thus applying Equation 2 yields

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1. \tag{3}$$

Two-qubit state is the quantum states executing on a pair of qubits, which can be calculated by the tensor product of two single-qubit states, *e.g.*, $|00\rangle = |0\rangle \otimes |0\rangle$. Meanwhile, any two qubits can be in the state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, where the amplitudes α, β, γ, and δ satisfy:

$$\begin{aligned}\alpha = a + b\cdot i, \beta = c + d\cdot i,\\ \gamma = e + f\cdot i, \delta = g + h\cdot i.\end{aligned} \tag{4}$$

Similar to single-qubit states, since the probabilities of all possible qubits equal 1, the four amplitudes satisfy $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. By applying Equation 4 to the above constraint, we have

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 + |e|^2 + |f|^2 + |g|^2 + |h|^2 = 1. \tag{5}$$

3. Related Work

Our work is relevant to prior studies on graphical tools in quantum computing and the visual design of quantum states.

3.1. Graphical Tools in Quantum Computing

Researchers have attempted to visualize quantum circuits using graphical interfaces, including quantum circuit implementation and building graphical simulators for quantum circuits. Specifically, Paykin *et al.* [PRZ17] presented an interface to manipulate quantum circuits using a classical host approach. Zulehner *et al.* [ZW18] proposed a new graph-based approach for quantum simulators. Much prior work has also been studied to interpret quantum algorithms and workflow using a graphical approach. Ruan *et al.* [RWJ*22] introduced a real-time visualization system for noise awareness in quantum computers and compiled circuits. Tao *et al.* [TPCW17] propose a tool to assist users in understanding Shor's algorithms using a graphical interface. In addition, *Quirk* [Qui14] also provides a graphical interface to make users aware of the quantum circuit's behavior. The aforementioned interfaces focused on improving the interpretability of quantum circuits using multiple fundamental visualizations individually (*e.g.*, bar charts, circles in *Quirk* [Qui14]). However, this type of visualization cannot intuitively visualize the relationship between properties in quantum computing, making the users feel struggling to understand the changes happening in quantum states. Our work addressed this challenge by introducing a visual solution, assisting users in quickly understanding single-qubit and two-qubit states thoroughly.

3.2. Visual Design of Quantum States

Many prior studies have focused on visualizing quantum states using 3D and 2D visualizations. For 3D representation, a Block Sphere [Blo46] visualizes single-qubit states based on a 3D geometrical representation and is still the widely-used visual representation till now [FA09, HD04]. Some prior work focused on extending Bloch Spheres. For example, Altepeter *et al.* [AJMK09] extended *Bloch Sphere* using the remote-state preparation protocol. IBM also provides a Bloch Sphere-like design called Q-Sphere [ibm14] to represent multiple states in a single sphere. In addition, many researchers have studied how to represent quantum states using 2D shapes other than extending *Bloch Sphere*. For example, Galambos *et al.* [GI12] utilized fractal representation to visualize the multi-qubit qubit systems. Chernega *et al.* [CMM17] mapped the density matrix of the qubit onto the vertices of a triangle. While the prior studies provided different ways to portray the quantum states, none of the prior work focused on the representation of superposition, which reflects the probability of measuring each quantum state. We aim to visually correlate multiple properties with the probability of each quantum state, making the users understand the probability more comprehensively.

4. Informing The Design

In this section, we introduce our co-design process and the derived design requirements from it, which will inform our subsequent development of the visual design.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

4.1. Co-design Process

The primary goal of our co-design process is to collect the task abstractions faced by quantum computing users. Thus, following the guideline of design study [SMM12], we worked closely with five domain experts for over five months. First, we conducted preliminary interviews with five quantum computing experts, where we sought to understand the practical challenges and difficulties. Second, over the next four months, we revised our initial prototype iteratively according to the feedback collected from the expert test.

Participants. We invited five domain experts **P1-5** (5 males, $age_{mean} = 32.13$, $age_{sd} = 4.71$) to participate in our co-design process. Specifically, **P1** is a research scientist from Pacific Northwest National Laboratory in the U.S., **P2-5** are either professors or postdoc researchers from three different universities in the U.S. Among them, the research direction of **P1-2** and **P5** is Quantum Machine Learning, while **P3-4** are working on Quantum Systems. All the domain experts have an average of 6.3 years of research and development experience in quantum computing.

Preliminary Interview. Following the methodology proposed by Sedlmair et al. [SMM12], we began the preliminary interviews by performing one-on-one, semi-structured, hour-long interviews with Group 1 (**P1-3**), to collect current challenges the participants have when working on quantum computing. First, each participant was asked to describe the major issues and challenges they faced using the most popular visualization, *i.e.*, *Bloch Sphere*. Note that each participant in Group 1 used *Bloch Sphere* as the common tool in their daily routine tasks. Through this session, we collected a set of high-level requirements about the limitations of the *Bloch Sphere* representation. The aforementioned tasks last about 25 minutes. Then, we further asked every participant to describe the future requirements in a think-aloud manner, including those that can facilitate the visual analysis of quantum states. We summarized these initial requirements and expectations during this session. This session lasted about 30 minutes. We video-recorded and took notes for each interview and discussion. All requirements collected in this round were used to inform our initial prototype.

Expert Test. Over the next four months, we focused on iteratively testing the features of our initial prototype with Group 2 (**P4-5**). We designed and implemented the visual design according to the qualitative requirements collected from Group 1 and released it as an online interface, which allows Group 2 to access and use it. Specifically, we began the expert test by briefly introducing the session's purpose and describing the initial prototype's functionality. Each participant was encouraged to use our design when performing their daily domain tasks in quantum computing. We collected their feedback by holding iterative meetings with the participants every two weeks, which were open and loosely structured to ensure they could express anything they thought of. We further tuned the several design alternatives (as shown in Fig. 2) accordingly to guarantee our design meets all these practical needs.

4.2. Design Requirements

We summarized all design requirements from the co-design process. We reported six refined requirements and categorized them into **functionality** and **usability**.

For the functionality, participants reported three major requirements to facilitate the observation of arbitrary quantum states:

R1 Visualize two-qubit states in addition to one-qubit states. All participants (**P1-5**) reported a strong need to support the visualization of entanglement for two qubits. Thus, the basis for two-qubit entanglement representation is how to represent a two-qubit state. **P3** also suggested that it would be helpful to represent single-qubit and two-qubit states in the same set of visualization other than using two different forms of approaches.

R2 Provide an intuitive representation for the probability distribution of different states. All participants (**P1-5**) suggested that visually reflecting the probability distribution is essential for the quantum state exploration as the probabilities of different states explicitly reveal the *superposition*, which is the fundamental ingredient of quantum computing. Also, they preferred a visualization that can naturally show the probabilities with visual elements without any time-consuming manual calculation.

R3 Inform users of the state vector of each quantum state. Four participants (**P1-3, P5**) confirmed that showing the state vector (*e.g.*, amplitude α and β) as-is will greatly benefit quantum computing users. **P5** also reported that forming the visualization with states vectors is better for any quantum computing users due to every visual element's intuitiveness.

For usability, participants focused on how to make full use of the visualization for domain users, which was summarized as follows:

R4 Visually correlate elements in quantum states. Three participants (**P1-2**, **P5**) emphasized that it is significant to represent the state probabilities naturally by the basic element (*e.g.*, state vectors) other than an additional individual visual channel, since visually correlating different individual elements by the quantum mechanics theory makes the evolution of quantum states more easy-to-understand.

R5 Display all visual elements with 2D shapes. Three participants (**P3-5**) confirmed that they prefer a 2D representation to 3D of a quantum state. For example, the *Bloch Sphere* - the visualization tool that quantum computing users use most - is a 3D sphere. They commented that it is tough to locate the point's location and read the rotation angles accurately, even though the interaction of dragging is supported in some tools.

R6 Make the visualization tool accessible for all quantum computing users. Even though four participants (**P1-2**, **P4-5**) gave positive feedback for the useful prototype interface to assist their routine tasks during the stage of expert tests, they also emphasized the necessity of making it publicly-available to benefit all quantum computing users. Considering the power and popularity of the web-based cloud quantum computing platforms (*e.g.*, IBM Quantum), an online web interface will be a good choice.

5. *VENUS*

Drawing from the iterative co-design process, we designed *VENUS*, a novel visualization approach that facilitates the observation of the single-qubit and two-qubit quantum states. To enhance accessibility, we implemented a web-based graphical interface to fit into routine tasks. Users can access *VENUS* interface via https://venus-interface.github.io/. In this section, we first

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

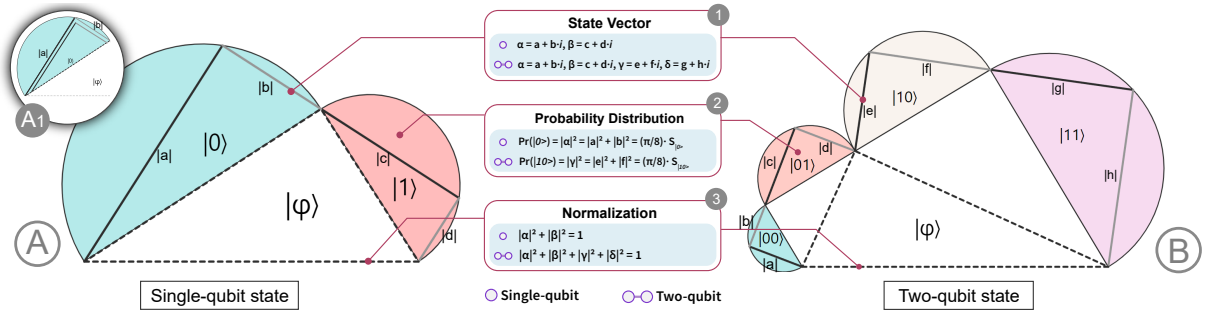


Figure 1: *The visual design of VENUS which supports single-qubit (A) and two-qubit (B) state representation based on the same visualization form.* ***Line segments*** *visualize the state vector, where the black line denotes the real part, and the grey line denotes the imaginary part based on Equation 1.* ***Semicircles****'s area indicates the probability of measuring the corresponding state based on Equation 2.* ***Triangle base****'s length consistently equals to 1, because it encodes based on the constraint of normalization (e.g., Equation 3).*

introduce the visual design of our visualization approach for single-qubit states and extend it to the form of two-qubit states.

5.1. Single-qubit State Representation

Informed by the design requirements introduced in Section 4.2, we propose a novel visual design for single-qubit state representation. As shown in Figure 1(A), we utilize three right triangles and two circumscribed semicircles to represent the quantum states.

State vector. According to the quantum computing theory, the state vector of a quantum state is the fundamental discipline to represent a quantum state. Thus, it is of great importance to encode the state vector in the visual design (**R3**). Specifically, we utilize each amplitude's real and imaginary parts (*e.g.*, a and b of amplitude α) to indicate the state vector according to the quantum theory illustrated in Section 2.3. Based on Equation 1, we utilize two line segments to depict the amplitudes of the state vector as shown in Figure 1 1. Also, we use cyan and red right triangles to indicate the two amplitudes α and β of the state vector, respectively. We encode the absolute values of real parts (*i.e.*, a and c) and imaginary parts (*i.e.*, b and d) of each amplitude by the line segments in black and grey, respectively. We apply the double lines, as shown in Figure 1 A1, to reflect the negative real and imaginary values. Note that if the number of the imaginary part is zero, the black line segment for the real part will coincide with the diameter of the semicircle.

Probability of the quantum states. According to the refined requirement, the probability of each state (*e.g.*, $|0\rangle$) is significant in revealing the superposition in quantum computing (**R2**). Based on Equation 2, we already know that the probability of a quantum state can be calculated by the absolute value of a and b. Meanwhile, for *VENUS*, according to the geometry of the visual design, the area of the semicircle circumscribed by the right triangle indicating the state vector can be calculated as follows:

$$S_{semicircle} = \frac{\pi}{8} \cdot (|a|^2 + |b|^2), \tag{6}$$

where a and b are the real and imaginary parts of α. Thus, building upon the above two equations, we can represent the probability of a specific quantum state by the area of the semicircle circumscribed by the right triangle since the area of the semicircle is proportional to the probability of the quantum state, as shown in Figure 1 2. Thus, it is apparent to visually analyze the two probabilities of $|0\rangle$ and $|1\rangle$ by the semicircle area based on the numerical state vectors without any manual calculation for the probability. Also, users are allowed to visually analyze how a certain state vector (*i.e.*, line segments) affects the corresponding probability.

Correlations between all elements. Through the co-design process, all participants pointed out that it is significant to visually link all various elements based on the normalization constraint, because this can highlight how state vector and corresponding probability affect each other (**R4**). As shown in Figure 1 3, we arrange all elements in quantum states into a systematical form of shapes. We arrange the visual channels of all elements in the quantum state into a set of right triangles with the base side length of 1 according to Equation 3. Building upon this rule, the bottom right triangle in white is used to link the other two right triangles in cyan and red for two states (*i.e.*, $|0\rangle$ and $|1\rangle$).

5.2. Two-qubit State Representation

From the co-design process, all domain experts strongly agreed that it would be much more beneficial if the visualization could support the two-qubit state representation (**R1**), which is one of the major limitations for the widely-used approach, *i.e.*, *Bloch Sphere*. Recall that the two-qubit state representation is the basis to make the two-qubit entanglement representation available. We extend from the approach for single-qubit state illustrated in Section 5.1 to unveil the mask of the two-qubit quantum states.

First, according to **R3**, we intend to present the two-qubit state based on the state vector. As shown in Figure 1(B), we utilize four pairs of line segments to visualize α, β, γ and δ building upon Equation 4. Each pair of line segments consists of a line segment in black to represent the real part and a line segment in grey to indicate the imaginary part of the complex number amplitudes (Figure 1 2).

Second, the probability distribution of quantum states is also supported for *VENUS*'s two-qubit mode (**R2**). In this case, there are four possible quantum states, *i.e.*, $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. By

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

applying Equation 2, the probabilities of the above four quantum states are proportional to the area of the corresponding circumscribed semicircle. So we encode the four two-qubit quantum states by four semicircles colored blue, red, almond, and purple. In this way, we convert the probability distribution for the two-qubit entanglement case by the graphical area of the four semicircles.

Third, inspired by **R4**, we correlate all elements for entangled two-qubit states (*i.e.*, four quantum states' state vectors and corresponding probabilities) by the geometry of *VENUS* intrinsically. Compared to the single-qubit representation, we utilize three auxiliary triangles (the white triangles in Figure 1Ⓑ) to correlate all colored triangles for quantum state representation based on Equation 5. It is clear that the base length of the other two triangles equals the bottom triangle's side length, which is always 1.

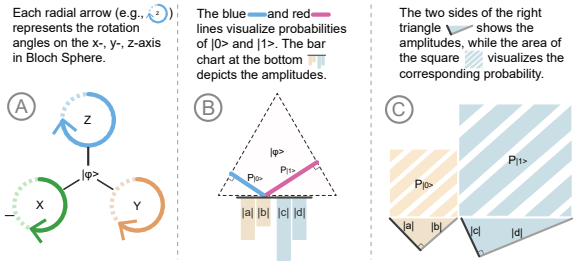


Figure 2: *The design alternatives of VENUS. (A) The visualization simply converts the three angles of the Bloch Sphere to 2D shapes. (B) A triangle-like design that visualizes the probabilities of two basic states (i.e., $|0\rangle$ and $|1\rangle$) using Viviani's theorem. (C) The design utilizes the square and equilateral triangle area to explicitly display the correlation between the probability and amplitudes.*

5.3. Design Justification

We considered three design alternatives before we came up with the final design for quantum state representation, as shown in Figure 2. Specifically, Figure 2Ⓐ shows the three axes (*i.e.*, x-axis, y-axis, and z-axis) of *Bloch Sphere*, which simply converts the quantum states in *Bloch Sphere* representation to 2D shapes. However, this approach cannot display the state vector and probability distribution. Figure 2Ⓑ encodes the probabilities of the two single-qubit states (*i.e.*, $|0\rangle$ and $|1\rangle$) by the length of blue and red line segments within an equilateral triangle, whose sum is a constant (*e.g.*, the sum of two probabilities that equals 1) due to the geometrical principle of equilateral triangles, *i.e.*, *Viviani's theorem* [Kaw05]. However, the domain experts pointed out in our co-design process that this design could not support the two-qubit state scenarios. Figure 2Ⓒ leverages two or four combinations of a square and a right triangle to visualize single-qubit and two-qubit states, respectively, where the length of two sides of triangles indicates the real and imaginary parts of each vector amplitudes. However, it is difficult to analyze the relationship of all quantum states and how the state vector will affect the probability. Thus, we further proposed the final designs (Figure 1) that meet all the above requirements.

6. Case Study

We utilized two applications, *i.e.*, a two-class quantum classifier and *Grover's algorithm*, to evaluate *VENUS* for single-qubit and two-qubit state representation, respectively. The participants were asked to use a monitor with 1920 × 1080 resolution in advance.

6.1. Case Study I - Single-qubit Quantum Classifier

E4 employed *VENUS* to explore various quantum states at the different stages of a quantum classifier. Specifically, E4 utilized Iris datasets [Iri11] to train the quantum classifier with a quantum simulator. Following the design methodology [Pen11], E4 encoded a single qubit by applying the two features of the Iris dataset. According to the network architecture proposed by Stein *et al.* [SLM*21], E4 first implemented the classifier circuit (Figure 3) using TorchQuantum [Tor11], which can support a flexible output for quantum states at each stage of the classification.

Understanding the learning process of the quantum classifier. E4 was curious about how the learning process will act on the data points. *"Generally, I faced the density matrix of the gates and qubits only, so it will be interesting to check out the evolution of the quantum states visually."* Keeping this in mind, E4 first selected a data point from the validation sets, whose two features were 1.4595 and 0.6797, respectively. After the selection, E4 glanced at the quantum state output at the first epoch out of 100. As shown in Figure 3A1, it is clear that the probability of state 0 is slightly larger than state 1, as indicated by the area of the two semicircles. E4 was surprised that *VENUS* shows the state probability from the amplitude values directly without any calculation E4 then noticed that the label of the data point is 1 *"Due to Label 1, I expect to see when the semicircle on the right will be larger than the left one, and how the state vector affects this convergence process."* So E4 evenly output the results of the data point's quantum states in subsequent three epochs (*i.e.*, 25, 50, 100) to see the evolution of the quantum classifier. As shown in Figure 3A2, the area of the semicircles indicating the probability of state 0 and 1 are almost the same at Epoch 25, and then the semicircle area of state 1 is significantly larger than that of state 0 at Epoch 50, and almost remains the same at Epoch 100. E4 commented that it is apparent that the quantum classifier has converged around Epoch 50 for this data point. Meanwhile, E4 noticed that this probability change was caused by the evolution of the corresponding state vector indicated by the line segments within the semicircles. In addition, E4 found that the increase of the real part in state 1 mainly results in the larger area of the red semicircle. E4 then confirmed, *"These findings provide me with a good guideline for model tuning. For example, during the parameter initialization of the rotation gates, I will choose those rotation angles which could make the absolute value of real parts larger than the imaginary part for both state 0 and 1."*

Unveiling the mask of the prediction process of QNN. Building upon the trained quantum classifier, E4 attempted to perform the prediction process and inspect the propagation of data points. First, E4 mentioned that despite the importance of quantum data embedding, it is still hard for him to understand how the original decimal data points are transformed into abstract quantum states. E4 hinted that the whole prediction process consists of four gates,

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

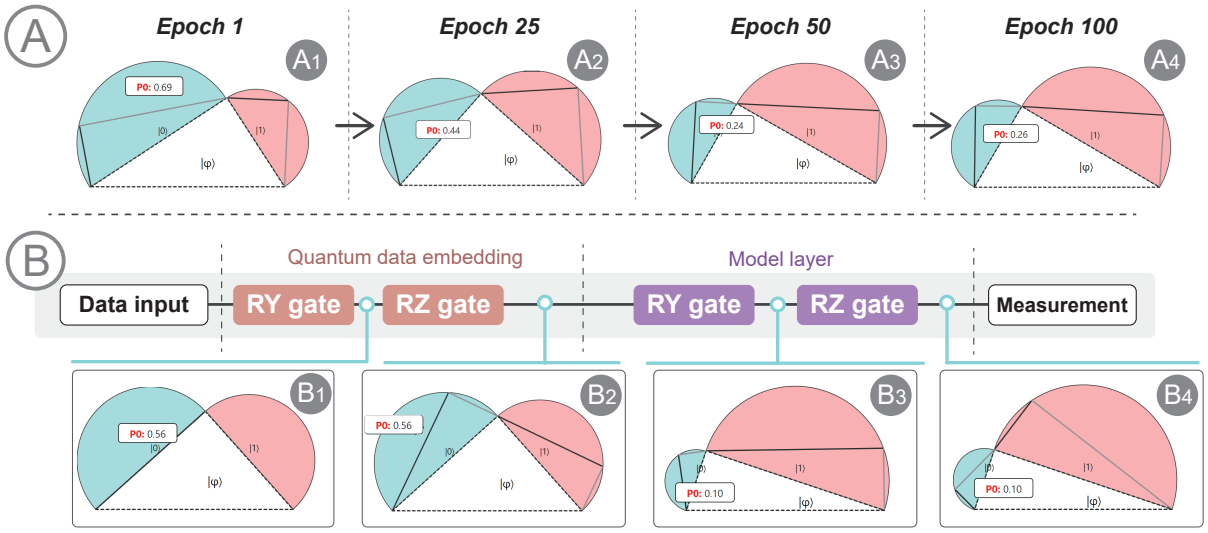


Figure 3: *The case for the single-qubit quantum classifier. (A) The learning process of the quantum classifier for the Iris dataset. The four charts indicate the evolution of a quantum classifier from Epoch 1 to Epoch 100, where the probability of state 1 is 31% at Epoch 1 and increased to 74% after 100 learning epochs. (B) The quantum states of an Iris data point in each step, which consists of two stages (i.e., quantum data embedding and the model layer). 'P0' indicates the probability the measurement result is 0.*

where the first two gates are used for quantum data embedding and the last two gates correspond to the trained classification model, as shown in Figure 3. Hence, E4 first randomly selected a data point and then visualized it after the first gate (*i.e.*, RY gate). E4 was surprised that the probabilities of the two states were around 0.50 (Figure 3B1): After the second quantum gate RZ gate (Figure 3B2), E4 further noticed that the line segments representing amplitudes coincide with the semicircles' diameters, while the amplitudes' imaginary parts convert to non-zero values. E4 explained this phenomenon, *"RZ is better to encode information because RZ is for the phase rotation, so this converting makes the input data more resilient to errors due to non-zero imaginary parts."* After exploring the two gates for quantum data embedding, E4 started to explore the two gates representing the classification model. He first inspected the quantum state after RY gate (Figure 3B3). E4 noticed that the probability of state 0 decreased from 0.56 to 0.10, as indicated by the semicircles' area, and almost remained the same after the last gate (Figure 3B4). *"This is mainly because the absolute values of amplitudes have increased and then cause the area of the probability of State 0 to increase"*, E4 commented.

6.2. Case Study II - Two-qubit Grover's Algorithm

E9's research interest lies primarily in *Grover's algorithm* [Gro96], a famous algorithm for the unstructured searching problem. Thus, E9 planned to explore the evolution of quantum states in *Grover's algorithm*. Following [Gro26], E9 implemented the circuit with one iteration and the target "winner" of state $|11\rangle$.

Revealing the insights of the hidden quantum states in Grover's algorithm. E9 was curious about the insights of the quantum states provided by *VENUS*, and expected to see how different modules interact with each other from a view of the functionality block other than individual quantum gates. Thus, E9 input the density matrix of quantum states into *VENUS*. After a glance at the results of the Hadamard gates (Figure 4A1), which is used to generate the superposition of qubits, E9 noticed that the probabilities of the four states ($|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$) are the same, as indicated by the white rectangle at the center and the same area of the four semicircles. *"It is clear that there is a uniform superposition at this stage because I found the probabilities are all the same. Besides, the imaginary parts are 0 for all quantum states since the line segments coincide with the semicircle."* E9 then exported the chart after the oracle process (Figure 4A2) that is used to flip the phase of the searched state. He quickly identified a phase flip indicated by the double line for state $|11\rangle$. Thus, E9 reported, *"Clearly, the quantum state $|11\rangle$ is the one we marked before the initialization due to the negative amplitude shown by the double line."* Building upon these findings of the pre-process before the iteration, E9 expected to see *"what the state "looks like" after the whole iteration stage."* Hence, E9 exported the visualization of the state (Figure 4A5) after the diffuser process, which is for the amplitude amplification of the target state. He quickly found that only a purple semicircle remained in the chart. *"To my surprise, all semicircles and triangles disappeared compared to the previous figure. This is probably because the iteration found the winner, whose probability is 1.00"*, E9 said. E9 commented, *"It is interesting to see the original two-qubit state (i.e., four semicircles) convert to only one semicircle, indicating the searching has converged."*

Performing a what-if analysis for multiple iterations. When using *VENUS* to explore the quantum state evolution of *Grover's algorithm*, E9 was also interested in exploring what would happen if another iteration was implemented on the circuit, as more iteration processes may lead to a better result. So he implemented one more iteration block and checked the results using *VENUS* (Fig-

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

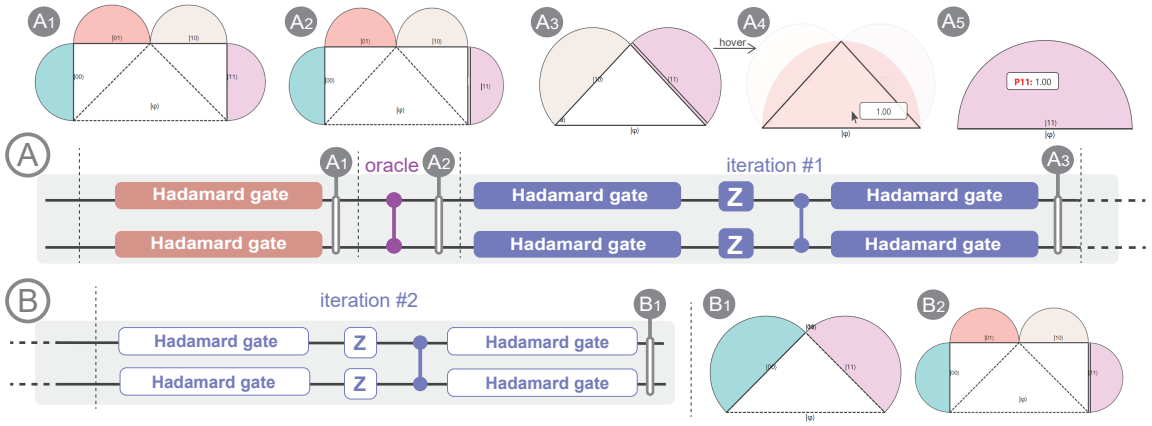


Figure 4: *The case for the two-qubit quantum algorithm, i.e., Grover's Algorithm. (A) The calculation process of Grover's Algorithm with one iteration, along with four consecutive quantum states and an interaction shown by VENUS. (B) The Grover's Algorithm with one more iteration appended after the original circuit, along with two quantum states representation.*

ure 4(B)). *"As I expected, the searching did not work with another iteration, because the four semicircles with an equal area of 0.25 (Figure 4(B2)) unlike the single output in the previous execution "*, E9 said. So, E9 attempted to explore the possible reasons. To this end, he compared all charts of each gate in the two iterations and quickly noticed that all states looked the same except those two after the last set of Hadamard gates in the respective iteration (Figure 4(A3) and Figure 4(B1)). Specifically, Figure 4(A3) shows that the two states (*i.e.*, $|10\rangle$ and $|11\rangle$) are with the same probability of 0.50 after the first Hadamard gate, while Figure 4(B1) indicated that the two states (*i.e.*, $|00\rangle$ and $|11\rangle$) are 0.50 at the same gate during the second iteration. Then, E9 hovered on the base triangle in Figure 4(A3) and the popped tooltip indicated that the probability is 1.00 when the first qubit's state is 1 (Figure 4(A4)). However, no tooltip popped out when he hovered over the second chart. Thus, E9 concluded that the failure of one more iteration is due to the differences between the two density matrices after the second last Hadamard gate. The last Hadamard gate can convert the state to the one with a probability of 1.00 (Figure 4(A4)) with the common first qubit's state with a probability of 0.50. *"This is because the Hadamard gate can output the unique value (i.e., 1.00) only if the operated state has the equal value on the first qubit's state."* E9 reported.

7. Expert Interview

We conducted in-depth expert interviews with 14 domain experts to evaluate the performance of *VENUS* for given tasks.

7.1. Participants and Apparatus

We invited 14 experts (2 females, 12 males) in quantum computing to join our well-designed interviews. Participants ($age_{mean} = 33.07$, $age_{sd} = 6.16$) were with an average of over five years of quantum computing research experience. Specifically, E4-8 are working on Quantum Machine Learning; E9-12 current research direction is Quantum Error Modeling; E13-14 are studying Quantum Chemistry, while the other three participants study Quantum Compiler (E1), Quantum Systems (E2), and Quantum Simulator (E3), respectively. Meanwhile, E3 and E12 are from the same research institute in the U.S., while others work in different educational institutions in the U.S. The above participants differ from the domain experts involved in the co-design process. All participants were asked to use a monitor with a resolution of 1920 $\times$ 1080 in advance.

7.2. Procedure

Before the formal study began, participants were asked questions about their demographics and research directions. We first introduced the goal and detailed procedures of the study, and then showcased an example to illustrate the interface and usage of *VENUS*. The above process lasted about 20 minutes.

After the introduction, they were expected to accomplish the predefined tasks described below in Section 7.3, which were about to observe the qubit state using the *VENUS* interface. Note that the application to perform the qubit state visualization was in line with the research direction of each participant so that we can evaluate the generalizability of *VENUS* for different domain tasks in quantum computing (*e.g.*, quantum algorithm, quantum machine learning). We recorded and took notes on each task and participants' interaction processes. The above process lasted about 45 minutes.

Upon task completion, all participants were then invited to participate in the post-study interview. Each participant was encouraged to describe the advantages and limitations in a think-aloud manner. Meanwhile, they were also encouraged to describe the issues of *Bloch Sphere* regarding these three aspects and how these issues affect their routine tasks. We invited every participant to rate the *VENUS* using a 7-Likert scale based on the questionnaire (Table 1). The above interview and rating procedures took 30 minutes.

7.3. Tasks

We designed the tasks to mimic the everyday tasks for analyzing the quantum states. To enhance the generalizability and make it not

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Table 1: *The questionnaire provided during the expert interview based on the tasks in the expert interview, i.e., Task 1 (**Q1-2**), Task 2 (**Q3-4**), Task 3 (**Q5-6**), and an overall feedback of VENUS (**Q7-9**).*

Q1	It is useful to show the quantum entanglement when observing quantum states.
Q2	It is easy to identify the entangled states via the visually correlated semicircles.
Q3	It is helpful to show the probability distribution.
Q4	It is intuitive to show probability distribution via the semicircle area.
Q5	It is informative to represent states via the state vectors.
Q6	It is easy to identify the state vectors via the line pairs within each semicircle.
Q7	The user interactions in the interface are useful and smooth.
Q8	The design can be integrated into the workflow well.
Q9	The 2D visual design is easy to view.

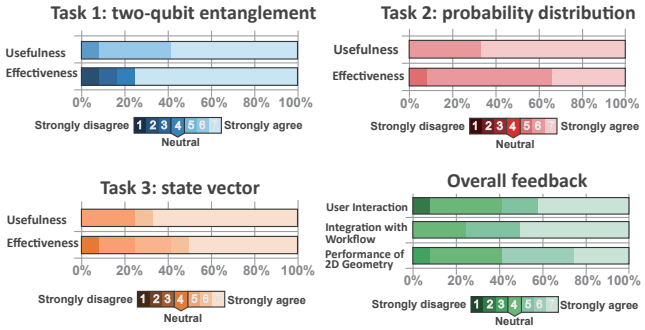


Figure 5: *The summary of the user feedback, which consists of the results of the three analytical tasks and the overall feedback.*

limited to one single application in quantum computing, we asked each participant to complete the tasks based on a typical example in their research directions (*e.g.*, variational quantum circuit for Quantum Machine Learning, QEC [LB13] algorithm for Quantum Error Modeling, or VQE [KMT*17] algorithm for Quantum Chemistry). We proposed three types of tasks as follows:

Task 1: Explore the two-qubit entanglement. Participants were asked to identify all two-qubit states in *VENUS*, describe the components in each two-qubit state, analyze the single-qubit state in the two-qubit quantum system states , and describe how this feature benefits their quantum state analysis.

Task 2: Observe the state probability distribution. Participants were asked to identify the probability of each quantum state and compare all quantum state probabilities via the corresponding semicircle area. User interactions displaying the exact probability amplitudes are allowed after the answer is given.

Task 3: Identify the state vector. Participants were asked to identify the real and imaginary parts of each state vector via the line segments on the right triangles and describe how the length of the line segments affects the semicircle area.

8. Interview Results

In this section, we report the summarized results, including the *Bloch Sphere*'s challenges, the responses for the three analytical tasks, and the suggestions provided by the participants. We reported the ratings from each participant in Figure 5.

8.1. Current Bloch Sphere Challenges

We invited all participants to participate in the feedback collection for *Bloch Sphere*, collecting challenges faced by the participants when attempting to perform quantum state observation using *Bloch Sphere*. We use this session to confirm our previous findings from the co-design process.

Challenge 1: Incapability of the two-qubit state representation. All participants agreed or strongly agreed that the inability to support two-qubit representation is a severe limitation for *Bloch Sphere*, they all agreed that it is unacceptable that *Bloch Sphere* cannot support multiple qubits' observation, which is the most critical property for quantum computing. E7 also reported that *Bloch Sphere* needs to be more scalable to allow the visualization of quantum entanglement.

Challenge 2: Non-intuitiveness of the state probability distribution. Most participants (11 out of 14) confirmed that *Bloch Sphere* could not intuitively visualize the probability of each possible quantum state. E13 mentioned that he has to manually calculate the final measurement from the point's location in *Bloch Sphere*. However, it is not a trivial task *"because the rotation angle in each axis of Bloch Sphere is hard to measure in a 3D model."*

Challenge 3: Lack of the state vector representation. Most participants (10 out of 14) agreed that the missing state vector in *Bloch Sphere* makes it challenging to understand. E1 commented, *"Despite the transparency of rotation gates for Bloch Sphere, but I prefer a more intuitive way to reflect the density matrix when designing circuits with quantum simulators."* E8 added that *Bloch Sphere* is unsuitable for educational purposes because of its complexity, especially for entry-level users.

8.2. Results of Three Analytical Tasks

We summarized all feedback regarding the three analytical tasks shown in Section 7.3.

Task 1: Two-qubit entanglement analysis. All participants agreed or strongly agreed that it was easy to support the two-qubit state observation using *VENUS* ($rating_{mean} = 6.01$, $rating_{sd} = 1.51$). Specifically, E3 believed that *VENUS*'s most practical usage is to *"make it possible for the entanglement display, which is the most important characteristic and needs for every quantum computing researcher."* E14, an expert in Quantum Chemistry, mentioned, *"my research is mainly focused on cutting the large quantum circuit; usually, the circuit has more than two qubits. This visualization of more qubits will help me to find the optimal cutting point more accurately."* Furthermore, as an expert on Quantum Error Modeling, E9 noted, *"I can see a great potential of VENUS. I think the quantum error correction (QEC) research can make good use of VENUS, for visualizing the encoding qubits and ancilla qubits separately."*

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Task 2: State probability observation. Most participants agreed that they could identify the probability of all possible quantum states quickly ($rating_{mean} = 6.60$, $rating_{sd} = 1.42$). E12 mentioned that the encoding area to visualize probability is *"easy and straightforward"*. E4 praised that, *"I like the idea of using the probability calculation equation to naturally visualize the probability distributin. I can directly check the probability without any manual calculation."* For the specific research directions in quantum computing, most participants (9 out of 12) highly appreciated *VENUS*'s usability and felt it could help them to handle their domain-specific tasks smoothly. For example, E4 confirmed that *"(for Quantum Machine Learning) VENUS can easily visualize the probability at breakpoints in debugging, which saves time in in-line debugging."*

Task 3: State vector exploration. Most participants agreed that it was intuitive to get a sense of the state vectors in detail using *VENUS* ($rating_{mean} = 5.85$, $rating_{sd} = 1.24$). Specifically, 8 out of 14 participants felt it helpful to show the amplitudes (*i.e.*, α and β) separately using multiple right triangles. E9 agreed that it provides a better view of the relationships between the probability distribution, especially the entanglements. Due to the accessibility of state vectors, E11 thought the visualizations of the real and imaginary parts of amplitudes were helpful *"for fresh starters in quantum computing."* Moreover, E2 confirmed the *VENUS*'s value in Quantum System, *"I can quickly compare the consequence of using different gates as it can fit into the debugger."* He also felt that *VENUS* could significantly reduce the requirements of developers to understand quantum physics, making it an excellent in-class teaching tool for non-physics background students.

8.3. Overall Feedback

From the participants' responses and ratings (*i.e.*, $rating_{mean} = 5.93$, $rating_{sd} = 1.03$), we summarize the overall feedback regarding the user interaction, the integration with the workflow, and the performance of 2D geometry. *1) User interactions.* Participants agreed that the user interactions of *VENUS* are valuable and smooth. They enjoyed the overall interactions during the study. Among them, participants gave highly positive feedback for the feature of the switching of qubit display orders. *2) Integration with the workflow.* Participants liked how *VENUS* can seamlessly fit into their specific domain tasks. Building upon the publicly-available platform, *VENUS* can benefit all quantum computing developers and researchers. *3) Performance of 2D geometry.* Participants agreed that using 2D shapes was more readable and apparent than 3D approaches. For instance, E6 pointed out that *VENUS* flattens the states and makes the perception of each characteristic more accurate than 3D views.

9. Discussion

In this section, we first summarize the lessons we learned during the development of *VENUS*, then we discuss the limitations of *VENUS*.

9.1. Lessons

During the above evaluation processes, all participants gave highly positive feedback for *VENUS*. Among all the responses, participants emphasized a strong need for visualization to fit into quantum computing regarding the complex quantum physics theory, non-transparency of quantum program process, and non-intuitive quantum computing properties (*i.e.*, quantum entanglement and superposition). The above challenges make it hard for novices and the general public to have a strong sense of quantum computing. Thus, the quantum computing community urgently needs visualization to aid the transparency and interpretability of quantum computing with its scientific educational capability.

9.2. Limitations

Our evaluation shows that *VENUS* can effectively facilitate quantum state observation. However, there are still some limitations.

Limited support for quantum noise visualization. *VENUS* can effectively visualize various quantum states in situations where noise analysis is not required, such as the design and debugging of quantum algorithms. We do not consider the noise analysis of *VENUS* because the design is built upon quantum simulators where the execution of quantum circuits is completely noise-free.

Scalability. *VENUS* currently targets visualizing the quantum states of one or two qubits. Compared with Block Sphere, it can effectively visualize the quantum entanglement, a significant step towards effective qubit state visualization confirmed by the participants. Also, E8 suggested enabling representation for more qubits by adding more triangles on top of *VENUS*. In the future, we will endeavor to extend the current design for more qubits.

Time-consuming input for state vectors. The update of *VENUS* is driven by the inputted number of amplitudes, which requires users to input the real and imaginary parts of amplitudes manually. Participants reported that it is inconvenient to type in the amplitude values. However, due to our contribution to a design study, we plan to address this limitation in the future. For example, as hinted by E4, converting from the popular visualization, *Bloch Sphere*, will also be helpful for users.

10. Conclusion

We present *VENUS*, a novel visualization approach for quantum state representation. Built upon the visualization of the state vectors, we utilized multiple coordinated 2D geometrical shapes to reflect the probability distribution of all quantum states, which provides an intuitive view of quantum superposition. Also, we applied the same form of visualization approaches to enable *VENUS* to support single-qubit and two-qubit state representations. We evaluated *VENUS* by conducting two case studies and in-depth expert interviews, and the results show that *VENUS* can effectively facilitate the visual analysis of single-qubit and two-qubit states.

In the future, we plan to enhance the data loading of quit states by making the user interface of *VENUS* able to read qubit states from quantum computing circuits automatically. Also, it will be interesting to investigate how visualization techniques can be applied to show the quantum states of more than two qubits.

Acknowledgment

This work was supported by the Lee Kong Chian Fellowship awarded to Dr. Yong Wang by SMU. Qiang Guan was supported by NSF 2212465, 2230111, 2217021 and 2238734.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

References

[AAB*19] ARUTE F., ARYA K., BABBUSH R., BACON D., BARDIN J. C., BARENDS R., BISWAS R., BOIXO S., BRANDAO F. G., BUELL D. A., ET AL.: Quantum supremacy using a programmable superconducting processor. *Nature 574*, 7779 (2019), 505–510. 1

[AJMK09] ALTEPETER J. B., JEFFREY E. R., MEDIC M., KUMAR P.: Multiple-qubit quantum state visualization. In *Proceedings of 2009 Conference on Lasers and Electro-Optics and 2009 Conference on Quantum Electronics and Laser Science Conference* (2009), IEEE, Optica Publishing Group, pp. 1–2. 3

[ARH*15] AMINI F., RUFIANGE S., HOSSAIN Z., VENTURA Q., IRANI P., MCGUFFIN M. J.: The impact of interactivity on comprehending 2d and 3d visualizations of movement data. *IEEE Transactions on Visualization and Computer Graphics 21*, 1 (2015), 122–135. doi:10.1109/TVCG.2014.2329308. 2

[BBF*20] BEER K., BONDARENKO D., FARRELLY T., OSBORNE T. J., SALZMANN R., SCHEIERMANN D., WOLF R.: Training deep quantum neural networks. *Nature Communications 11*, 1 (2020), 808. URL: https://doi.org/10.1038/s41467-020-14454-2, doi:10.1038/s41467-020-14454-2. 1

[BCLK*21] BHARTI K., CERVERA-LIERTA A., KYAW T. H., HAUG T., ALPERIN-LEA S., ANAND A., DEGROOTE M., HEIMONEN H., KOTTMANN J. S., MENKE T., ET AL.: Noisy intermediate-scale quantum (nisq) algorithms. *arXiv preprint arXiv:2101.08448* (2021). 1

[Blo46] BLOCH F.: Nuclear induction. *Physical Review 70*, 7-8 (1946), 460. 1, 3

[BSR21] BARDIN J. C., SLICHTER D. H., REILLY D. J.: Microwaves in quantum computing. *IEEE Journal of Microwaves 1*, 1 (2021), 403–427. 2

[CKJA*19] CÓRCOLES A. D., KANDALA A., JAVADI-ABHARI A., MCCLURE D. T., CROSS A. W., TEMME K., NATION P. D., STEFFEN M., GAMBETTA J. M.: Challenges and opportunities of near-term quantum computing systems. *arXiv preprint arXiv:1910.02894 108*, 8 (2019), 1338–1352. 1

[CMM17] CHERNEGA V. N., MAN'KO O. V., MAN'KO V. I.: Triangle geometry of the qubit state in the probability representation expressed in terms of the triada of malevich's squares. *Journal of Russian Laser Research 38*, 2 (2017), 141–149. 3

[FA09] FARGETTON R., ARRIGHI P.: The bloch representation of quantum states. *AIP Conference Proceedings 1110*, 1 (2009), 87–90. URL: https://aip.scitation.org/doi/abs/10.1063/1.3131387, doi:10.1063/1.3131387. 2, 3

[GI12] GALAMBOS M., IMRE S.: Visualizing the effects of measurements and logic gates on multi-qubit systems using fractal representation. *International Journal on Advances in Systems and Measurements Volume 5, Number 1 & 2, 2012* (2012). 3

[GJAE*20] GOKHALE P., JAVADI-ABHARI A., EARNEST N., SHI Y., CHONG F. T.: Optimized quantum compilation for near-term algorithms with openpulse. In *Proceedings of 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2020), IEEE, arXiv, pp. 186–200. 1

[Gom18] GOMES L.: Quantum computing: Both here and not here. *IEEE Spectrum 55*, 4 (2018), 42–47. 1

[GRL*03] GULDE S., RIEBE M., LANCASTER G., BECHER C., ESCHNER J., HÄFFNER H., SCHMIDT-KALER F., CHUANG I. L., BLATT R.: Implementation of the deutsch–jozsa algorithm on an ion-trap quantum computer. *Nature 421*, 6918 (2003), 48–50. 1

[Gro96] GROVER L. K.: A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1996), Association for Computing Machinery, pp. 212–219. 7

[Gro26] Grover' algorithm. https://qiskit.org/textbook/ch-algorithms/grover.html, Access Date: 2022-8-26. 7

[Hay08] HAYWARD M.: Quantum computing and shor's algorithm. *Sydney: Macquarie University Mathematics Department* (2008). 1

[HCS*20] HASSIJA V., CHAMOLA V., SAXENA V., CHANANA V., PARASHARI P., MUMTAZ S., GUIZANI M.: Present landscape of quantum computing. *IET Quantum Communication 1*, 2 (2020), 42–48. 1

[HD04] HAVEL T. F., DORAN C. J.: A bloch-sphere-type model for two qubits in the geometric algebra of a 6d euclidean vector space. In *Quantum Information and Computation II* (2004), vol. 5436, SPIE, SPIE, pp. 93–106. 2, 3

[Hey99] HEY T. A. J. G.: Quantum computing: an introduction. *Computing & Control Engineering Journal 10* (1999), 105–112. 2

[HIP*21] HUGHES C., ISAACSON J., PERRY A., SUN R., TURNER J.: *Quantum Computing for the Quantum Curious*. 01 2021. doi:10.1007/978-3-030-61601-4. 2

[ibm14] Ibm quantum. https://quantum-computing.ibm.com/, Access Date: 2022-7-14. 2, 3

[Iri11] Iris dataset. https://archive.ics.uci.edu/ml/datasets/iris, Access Date: 2022-8-11. 6

[Kaw05] KAWASAKI K.-I.: Proof without words: Viviani's theorem. *Mathematics Magazine 78*, 3 (2005), 213. 6

[KMT*17] KANDALA A., MEZZACAPO A., TEMME K., TAKITA M., BRINK M., CHOW J. M., GAMBETTA J. M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature 549*, 7671 (2017), 242–246. 9

[LB13] LIDAR D. A., BRUN T. A.: *Quantum error correction*. Cambridge university press, 2013. 9

[Mun09] MUNZNER T.: A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 921–928. 2

[MV17] MÖLLER M., VUIK C.: On the impact of quantum computing technology on future developments in high-performance scientific computing. *Ethics and Information Technology 19*, 4 (2017), 253–269. 1

[Nar99] NARAYANAN A.: Quantum computing for beginners. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (1999), vol. 3, pp. 2231–2238 Vol. 3. doi:10.1109/CEC.1999.785552. 2

[Pen11] Pennylane examples. https://pennylane.ai/qml/demos/tutorial_variational_classifier.html, Access Date: 2022-8-11. 6

[PRZ17] PAYKIN J., RAND R., ZDANCEWIC S.: Qwire: A core language for quantum circuits. *ACM SIGPLAN Notices 52*, 1 (2017), 846–858. 3

[Qui14] Quirk. https://github.com/Strilanc/Quirk, Access Date: 2022-7-14. 3

[RP00] RIEFFEL E., POLAK W.: An introduction to quantum computing for non-physicists. *ACM Comput. Surv. 32* (sep 2000), 300–335. URL: https://doi.org/10.1145/367701.367709, doi:10.1145/367701.367709. 2, 3

[RP11] RIEFFEL E. G., POLAK W. H.: *Quantum computing: A gentle introduction*. MIT Press, 2011. 1

[RWJ*22] RUAN S., WANG Y., JIANG W., MAO Y., GUAN Q.: Vacsen: A visualization approach for noise awareness in quantum computing. *arXiv preprint arXiv:2207.14135* (2022). 3

[SAP*21] SAKI A. A., ALAM M., PHALAK K., SURESH A., TOPALOGLU R. O., GHOSH S.: A survey and tutorial on security and resilience of quantum computing. In *Proceedings of the 2021 IEEE European Test Symposium (ETS)* (2021), IEEE, pp. 1–10. 1

[SLM*21] STEIN S. A., L'ABBATE R., MU W., LIU Y., BAHERI B., MAO Y., QIANG G., LI A., FANG B.: A hybrid system for learning classical data in quantum states. In *Proceedings of the 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)* (2021), pp. 1–7. doi:10.1109/IPCCC51483.2021.9679430. 1, 6

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

[SMM12] SEDLMAIR M., MEYER M., MUNZNER T.: Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2431–2440. 4

[Ste98] STEANE A.: Quantum computing. *Reports on Progress in Physics 61*, 2 (1998), 117. 1

[TCCG20] TACCHINO F., CHIESA A., CARRETTA S., GERACE D.: Quantum computers as universal quantum simulators: State-of-the-art and perspectives. *Advanced Quantum Technologies 3*, 3 (2020), 1900052. 1

[TKAM05] TORY M., KIRKPATRICK A. E., ATKINS M. S., MOLLER T.: Visualization task performance with 2d, 3d, and combination displays. *IEEE Transactions on Visualization and Computer Graphics 12*, 1 (2005), 2–13. 2

[Tor11] Torchquantum. https://github.com/mit-han-lab/torchquantum, Access Date: 2022-8-11. 6

[TPCW17] TAO Z., PAN Y., CHEN A., WANG L.: Shorvis: A comprehensive case study of quantum computing visualization. In *Proceedings of the 2017 International Conference on Virtual Reality and Visualization (ICVRV)* (2017), IEEE, pp. 360–365. 3

[TQ19] TANNU S. S., QURESHI M. K.: Not all qubits are created equal: A case for variability-aware policies for nisq-era quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2019), Association for Computing Machinery, pp. 987–999. 1

[Wie14] WIE C.-R.: Bloch sphere model for two-qubit pure states. *arXiv preprint arXiv:1403.8069* (2014). 2

[Wie20] WIE C.-R.: Two-qubit bloch sphere. *Physics 2*, 3 (2020), 383–396. 2

[Wil21] WILLIAMS M. M.: *QCVis: A quantum circuit visualization and education platform for novices*. PhD thesis, Harvard University, 2021. 2

[ZHVG20] ZABLE A., HOLLENBERG L., VELLOSO E., GONCALVES J.: Investigating immersive virtual reality as an educational tool for quantum computing. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2020), Association for Computing Machinery, pp. 1–11. 2

[ZW18] ZULEHNER A., WILLE R.: Advanced simulation of quantum computations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38*, 5 (2018), 848–859. 3

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.