United States Patent Application Publication	20250265744
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	Lin; Chuan-En et al.

SYSTEMS AND METHODS FOR GENERATING CREATIVE SKETCHES USING MODELS GUIDED BY SKETCHES AND TEXT

Abstract

Systems, methods, and other embodiments described herein relate to generating images and sketches iteratively using models guided by sketches and text for a design. In one embodiment, a method includes generating an image from a sketched stroke and text inputted to a learning model. The method also includes segmenting the image to identify boundary information with a segmentation model and extracting edge information from the image using an edge model. The method also includes rendering an estimated sketch of the image by computing an intersection between the boundary information and the edge information.

Inventors: **Lin; Chuan-En (Pittsburgh, PA), Kang; Hyeonsu B. (Pittsburgh, PA), Martelaro; Nikolas A. (Pittsburgh, PA), Kittur; Aniket D. (Pittsburgh, PA), Chen; Yin-Ying (San Jose, CA), Hong; Matthew K. (Mountain View, CA)**

Applicant: **Toyota Research Institute, Inc.** (Los Altos, CA)

Family ID: **1000007817353**

Assignee: **Toyota Research Institute, Inc. (Los Altos, CA); Toyota Jidosha Kabushiki Kaisha (Toyota-shi Aichi-ken, JP); Carnegie Mellon University (Pittsburgh, PA)**

Appl. No.: **18/640519**

Filed: **April 19, 2024**

Related U.S. Application Data

us-provisional-application US 63556088 20240221

Publication Classification

Int. Cl.: **G06T11/20** (20060101); **G06T7/12** (20170101)

U.S. Cl.:

CPC **G06T11/203** (20130101); **G06T7/12** (20170101); G06T2207/10024 (20130101); G06T2210/21 (20130101); G06T2210/62 (20130101)

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATIONS [0001] This application claims the benefit of U.S. Provisional Application No. 63/556,088, filed on Feb. 21, 2024, which is herein incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] The subject matter described herein relates, in general, to generating images using sketches, and, more particularly, to generating images and sketches iteratively using models guided by sketches and text.

BACKGROUND

[0003] Systems for designing creative objects and products can use software tools having design engines that shorten development cycles. Designers can manipulate sketches through the design engines and collaborate with multi-discipline teams (e.g., sales) with the software tools. Nevertheless, the software tools still demand that designers perform arduous tasks during development, such as color selection, image searching, and contouring. As such, the benefit to design cycles has limits with the software tools, particularly involving complex products (e.g., vehicles) having multiple facets and parts.

[0004] In various implementations, systems use a text-to-image (T2I) model that is trained to synthesize images using text for design. For example, a T2I model generates shapes for a part that completes a vehicle guided with textual inputs. Notwithstanding, these systems encounter difficulties generating images that include detailed features described with text and inputted design parameters. As such, the designer may manually alter and manipulate generated designs according to the design parameters. The alteration increases design iterations and time, thereby reducing the efficiency benefits of the T2I model. Therefore, systems designing objects using models with text can lack details and limit design creativity.

SUMMARY

[0005] In one embodiment, example systems and methods relate to generating images and sketches iteratively using models guided by sketches and text for a design. In various implementations, systems implementing a text-to-image (T2I) model accelerate design cycles through converting textual ideas into visuals. As such, designers rely on T2I models to support and enhance creative tasks. Still, designers encounter difficulties with completing creative tasks despite advanced capabilities of the T2I model. For example, generating a vehicle part with T2I frustrates designers when the T2I poorly tracks a specific theme and outputs haphazard renders having a vehicle incorporating the vehicle part. Therefore, systems running a design engine with T2I can lack creativity and face limitations for generating images efficiently that meet design goals.

[0006] Therefore, in one embodiment, a drawing system inspires unique designs from various domains while reducing development through iterative sketching using models that are guided by sketches and text. In particular, the drawing system renders an estimated sketch of an image generated by a learning model as feedback for further enhancing design ideas. In one approach, the drawing system computes the estimated sketch through identifying an intersection between boundary information and edge information outputted by a data-driven model and an edge model, respectively. Furthermore, the drawing system also controls design over the ideation generation and concept exploration through modifying the estimated sketches and inputting additional text to guide the learning model. Accordingly, the drawing system produces design ideas from nature, architecture, fashion, etc., related to a product specified with text and a sketch as inputs and the estimated sketch.

[0007] In one embodiment, a drawing system for generating images and sketches iteratively using models guided by sketches and text for a design is disclosed. The drawing system includes a memory including instructions that, when executed by a processor, cause the processor to generate an image from a sketched stroke and text inputted to a learning model. The instructions also include instructions to segment the image to identify boundary information with a segmentation model and extract edge information from the image using an edge model. The instructions also include instructions to render an estimated sketch of the image by computing an intersection between the boundary information and the edge information.

[0008] In one embodiment, a non-transitory computer-readable medium for generating images and sketches iteratively using models guided by sketches and text for a design and including instructions that when executed by a processor cause the processor to perform one or more functions is disclosed. The instructions include instructions to generate an image from a sketched stroke and text inputted to a learning model. The instructions also include instructions to segment the image to identify boundary information with a segmentation model and extract edge information from the image using an edge model. The instructions also include instructions to render an estimated sketch of the image by computing an intersection between the boundary information and the edge information.

[0009] In one embodiment, a method for generating images and sketches iteratively using models guided by sketches and text for a design is disclosed. In one embodiment, the method includes generating an image from a sketched stroke and text inputted to a learning model. The method also includes segmenting the image to identify boundary information with a segmentation model and extracting edge information from the image using an edge model. The method also includes rendering an estimated sketch of the image by computing an intersection between the boundary information and the edge information.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate various systems, methods, and other embodiments of the disclosure. It will be appreciated that the illustrated element boundaries (e.g., boxes, groups of boxes, or other shapes) in the figures represent one embodiment of the boundaries. In some embodiments, one element may be designed as multiple elements or multiple elements may be designed as one element. In some embodiments, an element shown as an internal component of another element may be implemented as an external component and vice versa. Furthermore, elements may not be drawn to scale.

[0011] FIG. **1** illustrates one embodiment of a drawing system that generates images and sketches iteratively using models for design ideation including a feedback loop.

[0012] FIG. **2** illustrates one embodiment of a pipeline for the drawing system that generates an image using models that are guided by inputted sketches, estimated sketches, and text.

[0013] FIG. **3** illustrates one embodiment of the drawing system having a model rendering an estimated sketch from the image generated with a learning model.

[0014] FIG. **4** illustrates an example of a user interface (UI) that can modify the image generated with the learning model rendered as a sketch.

[0015] FIG. **5** illustrates one embodiment of a method that is associated with identifying boundary and edge information about the image for rendering the estimated sketch using models.

DETAILED DESCRIPTION

[0016] Systems, methods, and other embodiments associated with generating images and sketches iteratively using models guided by sketches and text for a design are disclosed herein. In various implementations, a drawing tool that generates ideas with a text-to-image (T2I) model encounters difficulties with interpreting abstract themes. For example, the drawing tool generates a tank for the input “a protected vehicle” as the abstract theme, thereby producing unsatisfactory outcomes. Furthermore, drawing tools can lack effective feedback features for design iteration. As such, the drawing tool having the T2I model exhibits an experience that is one-way and random.

[0017] Therefore, in one embodiment, a drawing system generates product designs through sketching and text with a feedback loop that improves iterative ideation. In particular, the feedback loop includes a model that transforms an image generated by a T2I model (e.g., a large language model (LLM)) and a generative model into a sketch that is modifiable. In this way, the drawing system allows continuous iterations that enhance and improve generative designs. In one approach, the drawing system segments an image generated by a learning model using text and a sketched stroke and identifies boundary information. Through segmentation, the drawing system can generate natural borders about the image using a segmentation map that highlights key areas for rendering a high-definition sketch. Furthermore, an edge model extracts edge information from the image for locating key areas within a generative image. Subsequently, the drawing system renders an estimated sketch of the image by computing an intersection between the boundary information and the edge information. In this way, the natural borders of the image are emphasized through removing texture and redundant lines with the intersection representing areas within the image having similar visual data. Accordingly, the drawing system improves rapid design development by generating images iteratively using models guided by sketches and text and a feedback loop rendering realistic sketches of generative images.

[0018] It will be appreciated that for simplicity and clarity of illustration, where appropriate, reference numerals have been repeated among the different figures to indicate corresponding or analogous elements. In addition, the discussion outlines numerous specific details to provide a thorough understanding of the embodiments described herein. Those of skill in the art, however, will understand that the embodiments described herein may be practiced using various combinations of these elements.

[0019] FIG. **1** illustrates one embodiment of a drawing system **100** that generates images and sketches iteratively using machine learning (ML) models for design ideation including a feedback loop. The drawing system **100** is shown as including a processor(s) **110** that the drawing system **100** may access through a data bus or another communication path. In one embodiment, the drawing system **100** includes a memory **120** that stores a design module **130**. The memory **120** is a random-access memory (RAM), a read-only memory (ROM), a hard-disk drive, a flash memory, or other suitable memory for storing the design module **130**. The design module **130** is, for example, computer-readable instructions that when executed by the processor(s) **110** cause the processor(s) **110** to perform the various functions disclosed herein.

[0020] Moreover, in one embodiment, the drawing system **100** includes a data store **140**. In one embodiment, the data store **140** is a database. The database is, in one embodiment, an electronic data structure stored in the memory **120** or another data store and that is configured with routines that can be executed by the processor(s) **110** for analyzing stored data, providing stored data, organizing stored data, and so on. Thus, in one embodiment, the data store **140** stores data used by the design module **130** in executing various functions. In one embodiment, the data store **140** further includes the edge information **150** and the boundary information **160**. The edge information **150** can include soft edges detected using an edge model about an image generated outputted by learning models. Here, a soft edge can include structural lines having varying thicknesses, opacities, and transparency levels. Furthermore, the drawing system **100** can derive the boundary information **160** by estimating a segmentation map using a segmentation model. As explained below, natural borders for colored parts defined by the segmentation map can be utilized to derive the boundary information **160**.

[0021] Moreover, the drawing system **100** as illustrated in FIG. **2** is generally an abstracted form. The drawing system **100** and the design module **130**, in one embodiment, also include instructions that cause the processor(s) **110** to generate an image from a sketched stroke and text inputted to a learning model. Here, the learning model may have a pipeline including a LLM and a ML model (e.g., a control network (controlnet)) that generates the image. In one approach, the drawing system **100** segments the image to identify boundary information with a segmentation model (e.g., a data-driven model, a neural network, etc.) and extracts edge information from the image using an edge model. As further explained below, the design module **130** can render an estimated sketch of the image by computing an intersection between the boundary information **160** and the edge information **150**.

[0022] Now turning to FIG. **2**, one embodiment of a pipeline **200** for the drawing system **100** that generates an image using models that are guided by inputted sketches, estimated sketches that are modified, and text is illustrated. The pipeline **200** references one or more models in FIG. **2**. A model can include one or more networks, subnetworks, physical models, data-driven models, mathematical models, and so on as understood by those having ordinary skill in the art. In particular, a model sketch-to-design **210** converts a sketched stroke and text inputted about a design concept and generates product designs. For example, a designer inputs a product category (e.g., a vehicle) and a design concept that is abstract (e.g., protective) as the text along with the sketched stroke. A LLM within the sketch-to-design **210** can process the text to generate analogical inspirations for the design concept. An analogical inspiration may encompass a rapid identification of relevant connections from diverse domains (e.g., nature, architecture, fashion, etc.) for inputs. In particular, the text can prompt the LLM to detail a design principle for a subject. For instance, the prompt can be "Describe the key design principles in <subject> design in a brief description." The prompt also may be context for the LLM to generate well-thought-out inspirations.

[0023] Given the design principles, the drawing system **100** prompts the LLM to generate inspirations from diverse domains (e.g., nature, art, architecture, fashion, etc.). For instance, the prompt is that you are a <subject> designer and the design principles in <subject> design is from <design principles> associated with initial generative processing. The LLM then ideates inspirations for <subject> design to convey sensible <concept> from domains such as nature, art, architecture, fashion, etc. Furthermore, the drawing system **100** can specify formatting for an answer from the LLM. For example, a format is a bullet-pointed list (e.g., five items). As additional enhancements of the design concept through iterations, a designer may select a design idea recommended and continue branching out to explore further inspirations.

[0024] Upon selecting a design idea outputted by the LLM, a ML model within the sketch-to-design **210** receives the sketched stroke. For example, the sketched stroke is a single line captured by a canvas on a user interface. Here, the ML model may be a controlnet that receives the design idea as text and the sketched stroke for rendering a generative image **215**, such as with stable diffusion. In various implementations, stable diffusion is a model that includes a variational autoencoder (VAE) for compressing an image from a pixel space to a latent space having reduced dimensionality, thereby capturing improved semantics. The model can iteratively apply Gaussian noise to the compressed latent representation during forward diffusion. Furthermore, a U-Net denoises the output from forward diffusion backwards and derives a latent representation accordingly. The VAE may decode and generate a final image by converting the latent representation back into the pixel space.

[0025] Additionally, the drawing system **100** can continue guiding design ideas using additional strokes inputted and modifying sketches estimated with a model design-to-sketch **230**. A new design is generated with an additional stroke, thereby making the creation process iterative. The drawing system **100** also focuses on sketching rather than engineering text prompts that improves creativity and reduces development time. In one approach, the drawing system **100** maintains an initial seed between generations for consistency and image generation that is rapid. Here, a seed may be additional text, words, images, image descriptions, etc. that is inputted to the sketch-to-design **210**. In another approach, the drawing system **100** receives a remix command that changes to a different seed and the sketch-to-design **210** generates a varied image accordingly. This process can include remixing text to generate varied forms of the image from randomizing seeds associated with design parameters. For example, the design parameters describe through text and numerals one of a body line, an exterior contour, scenery information, a product type, a product feel, a product perception, etc. Subsequently, the design-to-sketch **230** can generate a sketch from a generative image outputted by the sketch-to-design **210** and provide a feedback loop that is interactive and focused.

[0026] FIG. **3** illustrates one embodiment of the drawing system **100** having the design-to-sketch **230** rendering an estimated sketch from an image generated with a learning model. Here, designers and stakeholders can build on generative images by converting designs into sketch-style visuals. This provides a feedback loop for further design enhancements and mitigates limitations associated with textual prompts. An estimated sketch allows designers to visualize an underlay on a canvas like tracing paper. The estimated sketch also encourages inspirations from previously generated images and overcomes the challenge of starting with a blank canvas, especially with early designs. In this way, the drawing system **100** implements iterative sketching for design ideation through a sketch-to-design-to-sketch feedback loop.

[0027] Moreover, the design-to-sketch **230** includes a model for image segmentation **310** (e.g., a data-driven model) that executes semantic, instance, etc. segmentation on the generative image **215** outputted from the sketch-to-design **210** and estimates a segmentation map. Correspondingly, the drawing system **100** can identify the boundary information **160** from the segmentation map and form a skeleton drawing **330**. For instance, the segmentation map includes natural borders that are separated by defined parts (e.g., silhouettes, colored parts, grey-scale units, etc.) of the generative image **215**. The defined parts may be identified according to estimated classifications by the segmentation model for areas within the generative image **215**. For example, the image segmentation **310** is trained with annotated data of a vehicle having colored parts differentiating between wheels and doors as semantic classes. Furthermore, the soft edges **320** extract and locate the edge information **150** that includes soft edges from the generative image **215** using an edge model. As previously explained, a soft edge can include structural lines having varying thicknesses, opacities, and transparency levels rather than common edges associated with the generative image **215**. In one approach, the edge model is a holistically-nested edge detection (HED) model that detects hard edges, soft edges, etc. from an image using hierarchical representations within a neural network (NN). In this way, the design-to-sketch **230** estimates a high-resolution and realistic sketch **350** using an intersection **340** operation between the edge information **150** and the boundary information **160**.

[0028] Regarding details about the intersection **340** operation, the design module **130** locates structural lines that dominate and natural borders of the generative image **215**, thereby eliminating redundant lines and unwanted texture from the realistic sketch **350**. Redundant lines detected by the soft edges **320** and found within the skeleton drawing **330** can reduce the clarity, definition, and reality of sketches. The intersection are areas within the generative image **215** having similar visual data that can increase focus on the structural lines. For example, a main silhouette of a finger within a generated object is thicker than the inner wrinkles. In one approach, the intersection **340** operation removes wrinkles, texture, and redundant patterns by functioning as “AND” computational logic for areas within the generated object having the main silhouette and edges. In this way, the drawing system **100** allows varying thicknesses and line opacity through processing with the soft edges **320** and manipulating the skeleton drawing **330** for creating realistic, natural, and defined aesthetics to the realistic sketch **350**.

[0029] Following the realistic sketch **350** being modified and refined with the sketch-to-design **210** and the design-to-sketch **230** that improves the generative image **215**, the design-to-real **220** generates visuals that demonstrate the generative image **215** within a realistic environment. In one approach, the design-to-real **220** is a pipeline that includes LLM prompting for describing a scene and scale associated with a subject. The design-to-real **220** then estimates depth (e.g., pixel depth) within the generative image **215** after scaling and manipulates the LLM output. In this way, the design-to-real **220** can plant the generative image **215** within a realistic scene that mimics natural lighting and shadows.

[0030] Turning now to FIG. **4**, an example of a user interface (UI) **400** that can modify an image generated with a learning model rendered as a sketch is illustrated. Here, the drawing system **100** can underlay an estimated sketch generated by the design-to-sketch **230** within a canvas **410** on the UI **400**. The estimated sketch can be rendered from an image generated by the sketch-to-design **210** using subject and concept inputs **420**, inspirations **430**, and a sketched stroke. A designer can modify the estimated sketch generated by the design-to-sketch **230** within the panel **440**. The sketch-to-design **210** processes the estimated sketch as modified to form a feedback loop and outputs the various designs **450** including placing images within realistic scenes. Therefore, ideation and creative design converge more rapidly with the drawing system **100** using the feedback loop.

[0031] Additional aspects of the drawing system **100** will be discussed in relation to FIG. **5**. FIG. **5** illustrates a flowchart of a method **500** that is associated with generating drawings iteratively using models guided by sketches and text for a design. Method **500** will be discussed from the perspective of the drawing system **100** of FIG. **1**. While method **500** is discussed in combination with the drawing system **100**, it should be appreciated that the method **500** is not limited to being implemented within the drawing system **100** but is instead one example of a system that may implement the method **500**.

[0032] At **510**, the drawing system **100** generates an image from a sketched stroke and text by a learning model. Here, the drawing system **100** can include a model sketch-to-design that converts a sketched stroke and text inputted about a design concept and generates product designs. For example, a designer inputs a product (e.g., a bumper) and a design concept that is abstract (e.g., hidden) as the text along with the sketched stroke. In one approach, the sketch-to-design includes a LLM that processes the text to generate analogical inspirations and ideas for the design concept. In particular, the text can prompt the LLM to detail design principles for a product including context for generating well-thought-out inspirations. The drawing system **100** can prompt the LLM to generate inspirations from diverse domains (e.g., nature, architecture, fashion, etc.) using the design principles.

[0033] Moreover, the LLM ideates inspirations for the product and concept. The design ideas are further enhanced through iterations until a designer selects a design idea that is recommended. Furthermore, a ML model (e.g., a controlnet) within the sketch-to-design receives the sketched stroke (e.g., a single line) and the design idea as text from a prompt and renders a generative image. In one approach, as explained more below, the drawing system **100** continues guiding the design idea using additional strokes inputted and modifying sketches estimated with a model design-to-sketch. The iterative process involves generating a new design with the additional strokes and modifying estimated sketches. In this way, the drawing system **100** fosters and develops design through sketching rather than mainly text prompts, thereby expanding creativity and accelerating design development.

[0034] At **520**, the drawing system **100** segments the image to identify the boundary information **160** with a segmentation model and extract the edge information **150** using an edge model. Here, the design-to-sketch includes another model for image segmentation (e.g., a data-driven model) that executes semantic, instance, etc. segmentation on the image from the sketch-to-design. The drawing system **100** can identify the boundary information **160** from a segmentation map outputted by the segmentation model and form a skeleton drawing. As previously explained, the segmentation map of the generative image can include natural borders that are separated using defined parts. For example, the defined parts are identified according to estimated classifications by the segmentation model for areas within the image.

[0035] Regarding extracting the edge information, the edge model can detect soft and other edges from the image generated with the sketch-to-design. As previously explained, a soft edge can include structural lines having varying thicknesses, opacities, and transparency rather than common and minor edges associated with the image. In one approach, the edge model is a HED model that detects hard edges, soft edges, etc. from an inputted image using hierarchical representations within a NN.

[0036] At **530**, the design module **130** renders an estimated sketch by computing an intersection between the boundary information **160** and edge information **150**. As previously explained, the intersection operation can allow the design module **130** to locate major lines and natural borders of the image. In this way, the operation eliminates redundant lines and certain texture, thereby increasing the clarity and definition of sketches. The intersection can be areas within the image that are segmented and exhibit similar visual data. In one approach, the intersection operation removes texture and redundant patterns by functioning as "AND" computational logic for areas within the image having a major silhouette and edges. As such, the drawing system **100** allows varying thicknesses and line opacity through processing the edge information **150** and the boundary information **160** derived with image segmentation that creates defined aesthetics to a sketch. Accordingly, the drawing system **100** improves ideation for designs through sketching and text guidance, thereby reducing design times and increasing designer satisfaction.

[0037] Detailed embodiments are disclosed herein. However, it is to be understood that the disclosed embodiments are intended as examples. Therefore, specific structural and functional details disclosed herein are not to be interpreted as limiting, but merely as a basis for the claims and as a representative basis for teaching one skilled in the art to variously employ the aspects herein in virtually any appropriately detailed structure. Furthermore, the terms and phrases used herein are not intended to be limiting but rather to provide an understandable description of possible implementations. Various embodiments are shown in FIGS. **1-5**, but the embodiments are not limited to the illustrated structure or application.

[0038] The flowcharts and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments. In this regard, a block in the flowcharts or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

[0039] The systems, components, and/or processes described above can be realized in hardware or a combination of hardware and software and can be realized in a centralized fashion in one processing system or in a distributed fashion where different elements are spread across several interconnected processing systems. Any kind of processing system or another apparatus adapted for carrying out the methods described herein is suited. A typical combination of hardware and software can be a processing system with computer-usable program code that, when being loaded and executed, controls the processing system such that it carries out the methods described herein.

[0040] The systems, components, and/or processes also can be embedded in a computer-readable storage, such as a computer program product or other data programs storage device, readable by a machine, tangibly embodying a program of instructions executable by the machine to perform methods and processes described herein. These elements also can be embedded in an application product which comprises the features enabling the implementation of the methods described herein and, which when loaded in a processing system, is able to carry out these methods.

[0041] Furthermore, arrangements described herein may take the form of a computer program product embodied in one or more computer-readable media having computer-readable program code embodied, e.g., stored, thereon. Any combination of one or more computer-readable media may be utilized. The computer-readable medium may be a computer-readable signal medium or a computer-readable storage medium. The phrase “computer-readable storage medium” means a non-transitory storage medium. A computer-readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer-readable storage medium would include the following: a portable computer diskette, a hard disk drive (HDD), a solid-state drive (SSD), a ROM, an EPROM or flash memory, a portable compact disc read-only memory (CD-ROM), a digital versatile disc (DVD), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer-readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0042] Generally, modules as used herein include routines, programs, objects, components, data structures, and so on that perform particular tasks or implement particular data types. In further aspects, a memory generally stores the noted modules. The memory associated with a module may be a buffer or cache embedded within a processor, a RAM, a ROM, a flash memory, or another suitable electronic storage medium. In still further aspects, a module as envisioned by the present disclosure is implemented as an ASIC, a hardware component of a system on a chip (SoC), as a programmable logic array (PLA), or as another suitable hardware component that is embedded with a defined configuration set (e.g., instructions) for performing the disclosed functions.

[0043] Program code embodied on a computer-readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber, cable, radio frequency (RF), etc., or any suitable combination of the foregoing. Computer program code for carrying out operations for aspects of the present arrangements may be written in any combination of one or more programming languages, including an object-oriented programming language such as Java™, Smalltalk.sup.M, C++, or the like and conventional procedural programming languages, such as the “C” programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer, or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0044] The terms “a” and “an,” as used herein, are defined as one or more than one. The term “plurality,” as used herein, is defined as two or more than two. The term “another,” as used herein, is defined as at least a second or more. The terms “including” and/or “having,” as used herein, are defined as comprising (i.e., open language). The phrase “at least one of . . . and . . . ” as used herein refers to and encompasses any and all combinations of one or more of the associated listed items. As an example, the phrase “at least one of A, B, and C” includes A, B, C, or any combination thereof (e.g., AB, AC, BC, or ABC).

[0045] Aspects herein can be embodied in other forms without departing from the spirit or essential attributes thereof. Accordingly, reference should be made to the following claims, rather than to the foregoing specification, as indicating the scope hereof.

Claims

1. A drawing system comprising: a memory storing instructions that, when executed by a processor, cause the processor to: generate an image from a sketched stroke and text inputted to a learning model; segment the image to identify boundary information with a segmentation model and extract edge information from the image using an edge model; and render an estimated sketch of the image by computing an intersection between the boundary information and the edge information.
2. The drawing system of claim 1, wherein the instructions to segment the image further include instructions to: estimate a segmentation map by the segmentation model; and draw natural borders for colored parts defined by the segmentation map to derive the boundary information.
3. The drawing system of claim 2, wherein the instructions to render the estimated sketch further include instructions to: remove texture and redundant lines for locating the natural borders according to the intersection between the boundary information and the edge information, wherein the intersection are areas within the image having visual data that is similar.
4. The drawing system of claim 2, wherein the natural borders separate the colored parts according to classifications identified by the segmentation map.
5. The drawing system of claim 1, wherein the edge information includes soft edges about the image, wherein the soft edges include structural lines having varying thicknesses, opacities, and transparency levels.
6. The drawing system of claim 1 further including instructions to: underlay the estimated sketch within a canvas on an interface; modify the estimated sketch within the interface; and receive by the learning model the estimated sketch as a feedback input.
7. The drawing system of claim 1, wherein the instructions to generate the image from the sketched stroke further include instructions to: remix the text to generate varied forms of the image from randomized seeds associated with design parameters, wherein the design parameters are associated with one of a body line, an exterior contour, scenery information, a product type, a product feel, and a product perception.
8. The drawing system of claim 1, wherein the instructions to generate the image from the sketched stroke further include instructions to: process the text by a large language model (LLM) of the learning model to output design ideas, wherein the text includes a product category and a design concept associated with the design ideas; and form the image by a control network (controlnet) of the learning model according to the design ideas and the sketched stroke.
9. A non-transitory computer-readable medium comprising: instructions that when executed by a processor cause the processor to: generate an image from a sketched stroke and text inputted to a learning model; segment the image to identify boundary information with a segmentation model and extract edge information from the image using an edge model; and render an estimated sketch of the image by computing an intersection between the boundary information and the edge information.
10. The non-transitory computer-readable medium of claim 9, wherein the instructions to segment the image further include instructions to: estimate a segmentation map by the segmentation model; and draw natural borders for colored parts defined by the segmentation map to derive the boundary information.
11. The non-transitory computer-readable medium of claim 10, wherein the instructions to render the estimated sketch further include instructions to: remove texture and redundant lines for locating the natural borders according to the intersection between the boundary information and the edge information, wherein the intersection are areas within the image having visual data that is similar.
12. The non-transitory computer-readable medium of claim 10, wherein the natural borders separate the colored parts according to classifications identified by the segmentation map.
13. A method comprising: generating an image from a sketched stroke and text inputted to a

learning model; segmenting the image to identify boundary information with a segmentation model and extracting edge information from the image using an edge model; and rendering an estimated sketch of the image by computing an intersection between the boundary information and the edge information.

14. The method of claim 13, wherein segmenting the image further includes: estimating a segmentation map by the segmentation model; and drawing natural borders for colored parts defined by the segmentation map to derive the boundary information.

15. The method of claim 14, wherein rendering the estimated sketch further includes: removing texture and redundant lines for locating the natural borders according to the intersection between the boundary information and the edge information, wherein the intersection are areas within the image having visual data that is similar.

16. The method of claim 14, wherein the natural borders separate the colored parts according to classifications identified by the segmentation map.

17. The method of claim 13, wherein the edge information includes soft edges about the image, wherein the soft edges include structural lines having varying thicknesses, opacities, and transparency levels.

18. The method of claim 13 further comprising: underlaying the estimated sketch within a canvas on an interface; modifying the estimated sketch within the interface; and receiving by the learning model the estimated sketch as a feedback input.

19. The method of claim 13, wherein generating the image from the sketched stroke further includes: remixing the text to generate varied forms of the image from randomized seeds associated with design parameters, wherein the design parameters are associated with one of a body line, an exterior contour, scenery information, a product type, a product feel, and a product perception.

20. The method of claim 13, wherein generating the image from the sketched stroke further includes: processing the text by a large language model (LLM) of the learning model to output design ideas, wherein the text includes a product category and a design concept associated with the design ideas; and forming the image by a control network (controlnet) of the learning model according to the design ideas and the sketched stroke.