



US 20250265752A1

(19) **United States**

(12) **Patent Application Publication**

**Sree Harsha et al.**

(10) **Pub. No.: US 2025/0265752 A1**

(43) **Pub. Date: Aug. 21, 2025**

(54) **DIGITAL VIDEO EDITING BASED ON A  
TARGET DIGITAL IMAGE**

(52) **U.S. Cl.**  
CPC ..... **G06T 11/60** (2013.01); **G06T 5/70**  
(2024.01)

(71) Applicant: **Adobe Inc.**, San Jose, CA (US)

(72) Inventors: **Sai Sree Harsha**, La Jolla, CA (US);  
**Dhwanit Agarwal**, San Jose, CA (US);  
**Ambareesh Revanur**, San Jose, CA  
(US); **Shradha Agrawal**, Milpitas, CA  
(US)

(73) Assignee: **Adobe Inc.**, San Jose, CA (US)

(21) Appl. No.: **18/583,067**

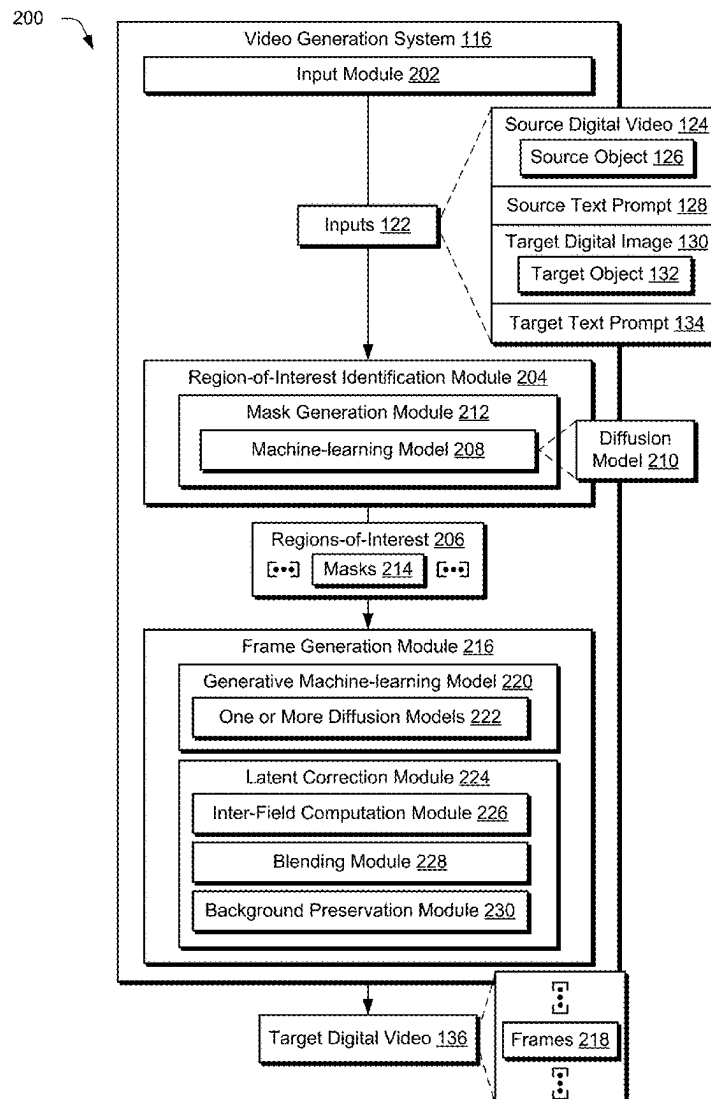
(22) Filed: **Feb. 21, 2024**

**Publication Classification**

(51) **Int. Cl.**  
**G06T 11/60** (2006.01)  
**G06T 5/70** (2024.01)

(57) **ABSTRACT**

Digital video editing techniques are described that are based on a target digital image. In one or more implementations, inputs are received. The inputs include a target text prompt, a target digital image depicting a target object, and a source digital video having a plurality of frames depicting a source object. Regions-of-interest are identified in the plurality of frames of the source digital video, respectively, based on the target text prompt and the target digital image using a machine-learning model, e.g., a diffusion model. A plurality of frames of a target digital video are generated as having the target object using a generative machine-learning model. The generating is based on the regions-of-interest, the target digital image, the source digital video, and a source text prompt describing the source digital video.



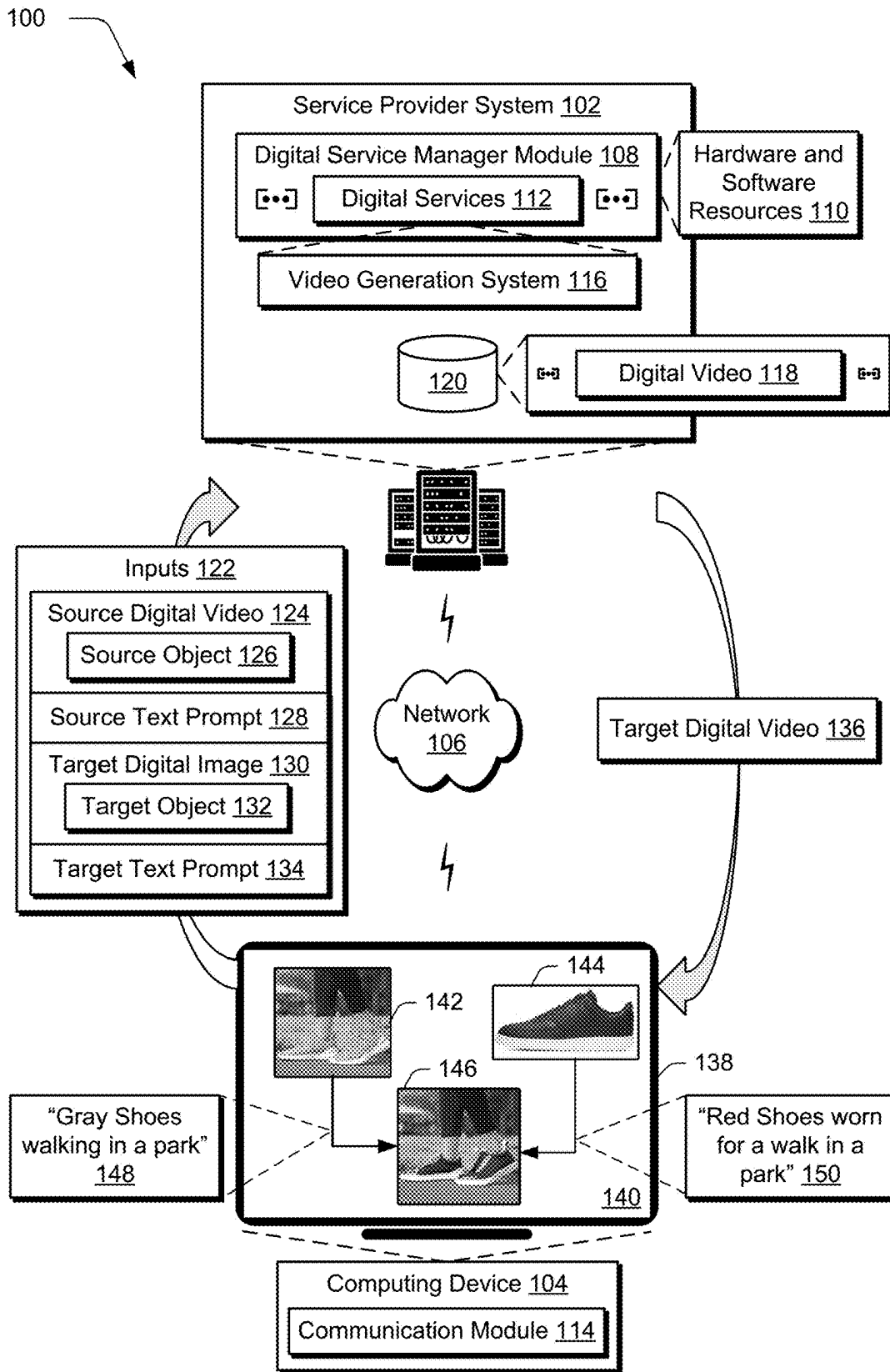


Fig. 1

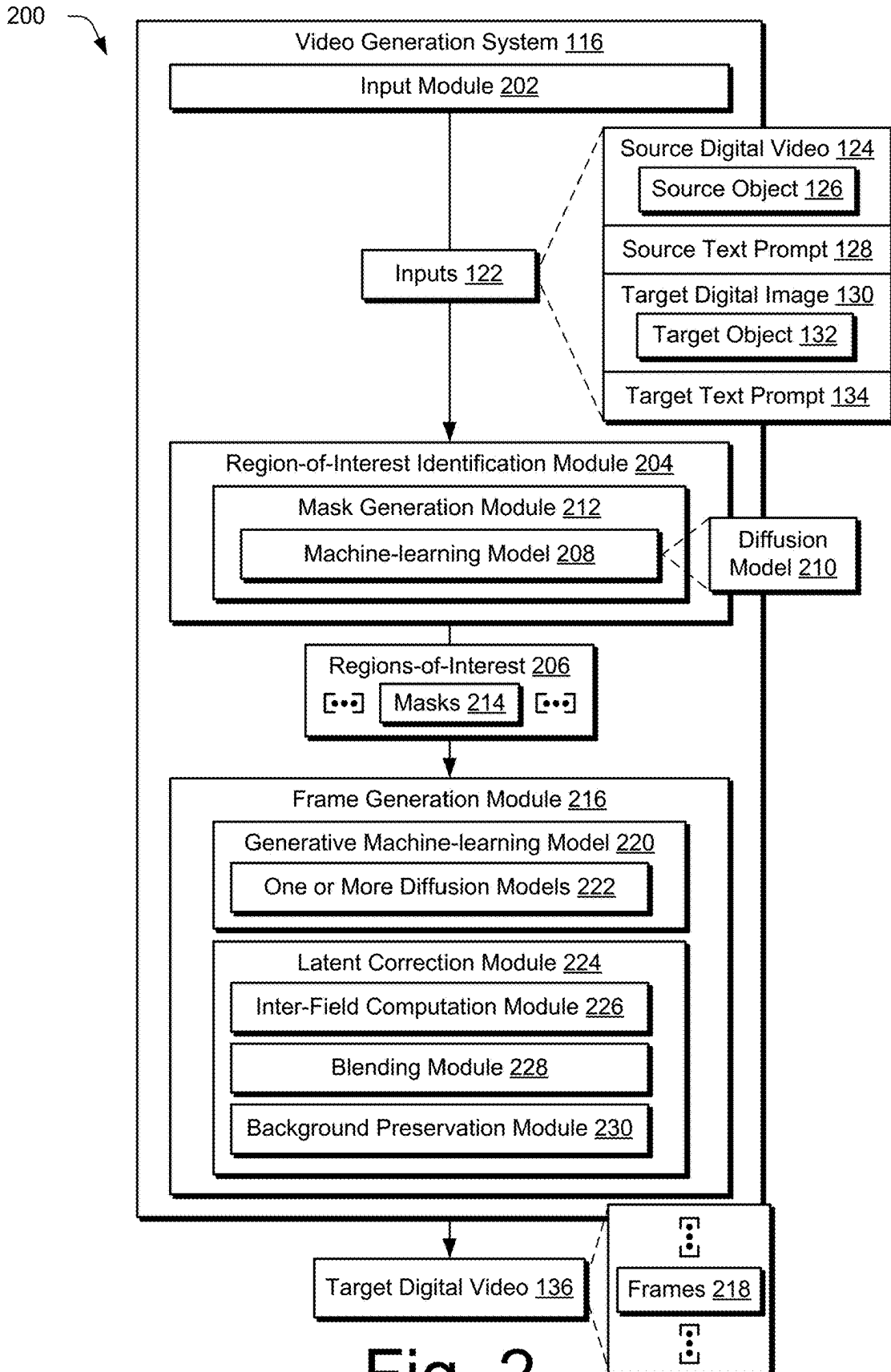


Fig. 2

300

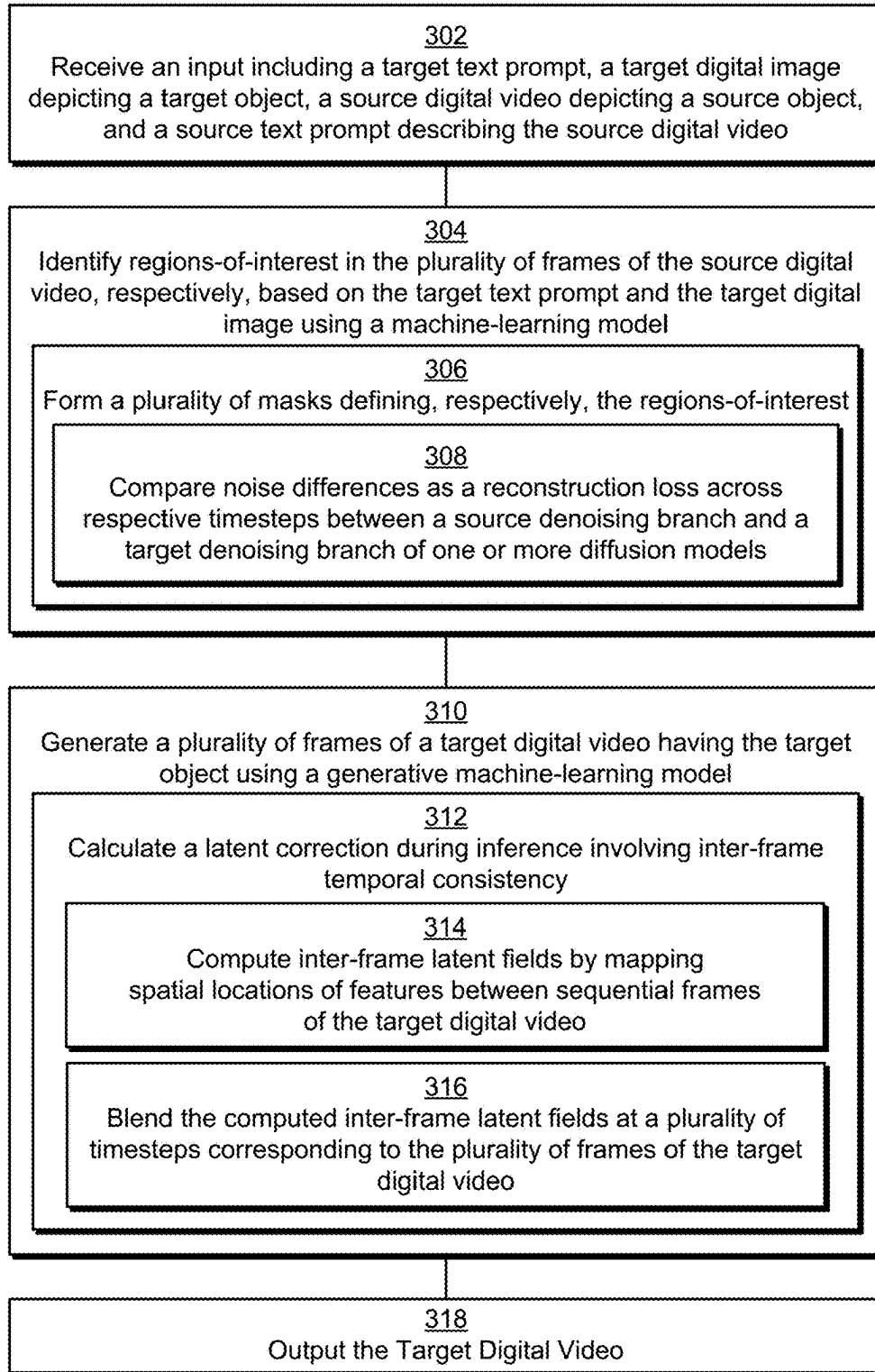


Fig. 3

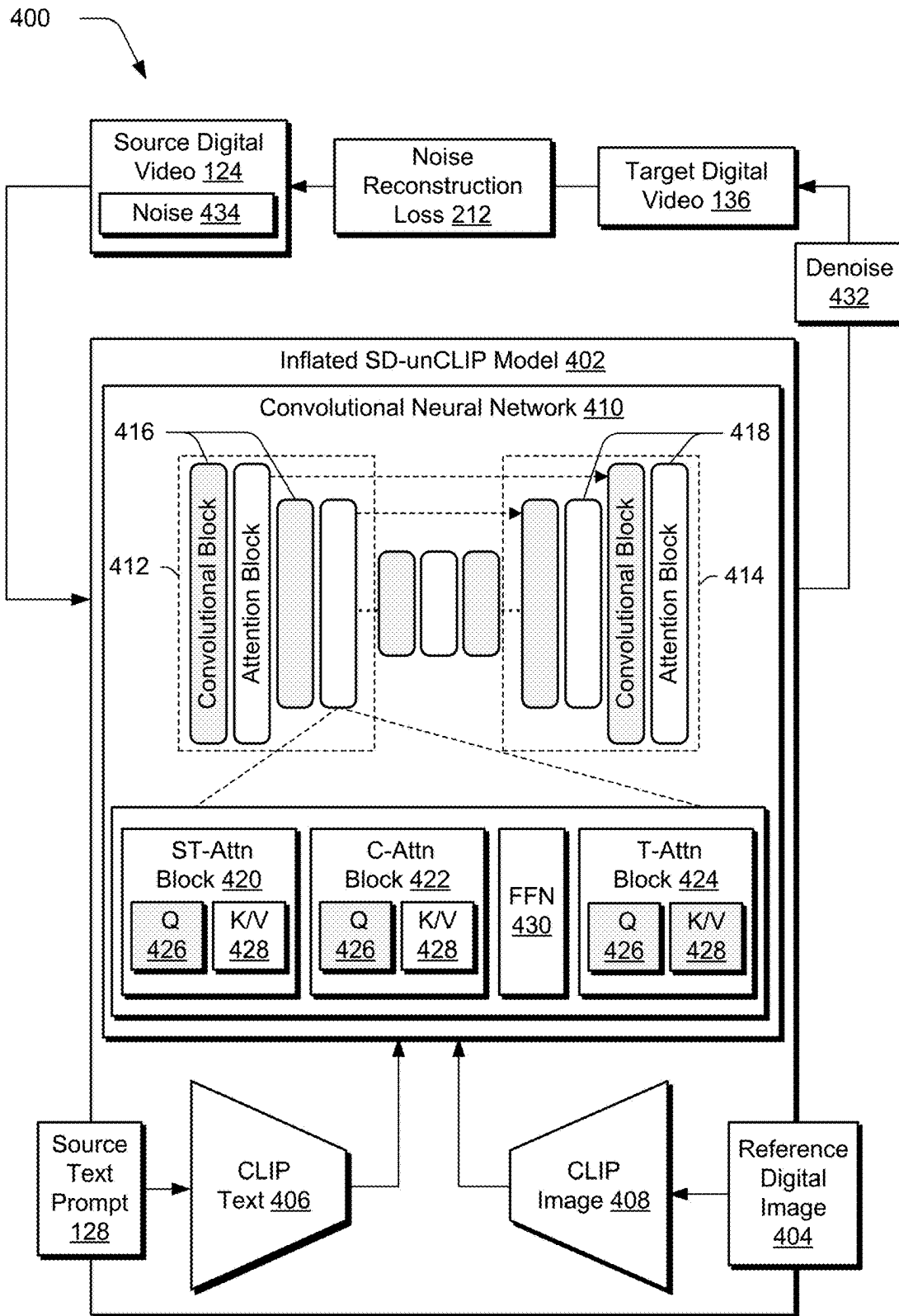


Fig. 4

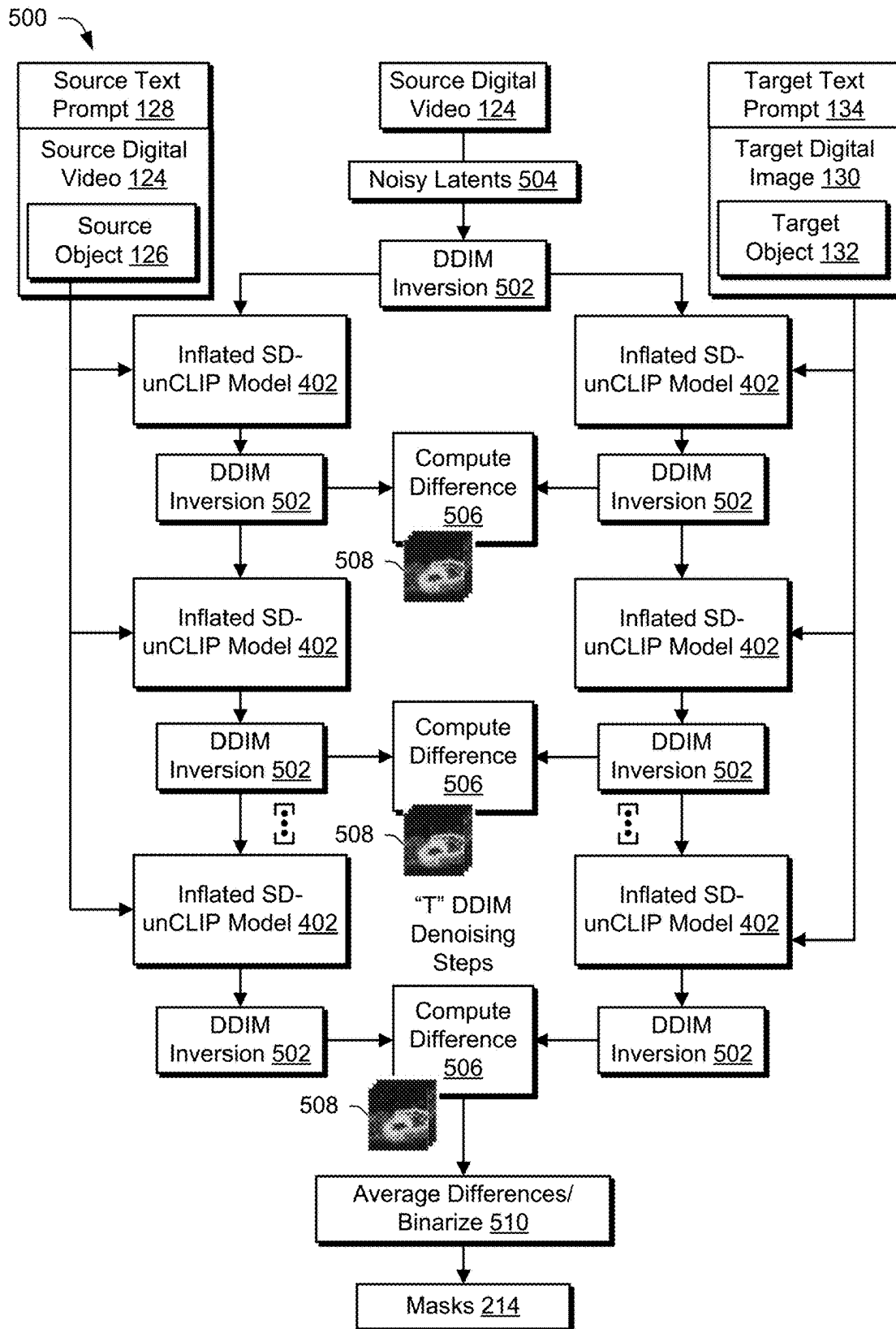


Fig. 5

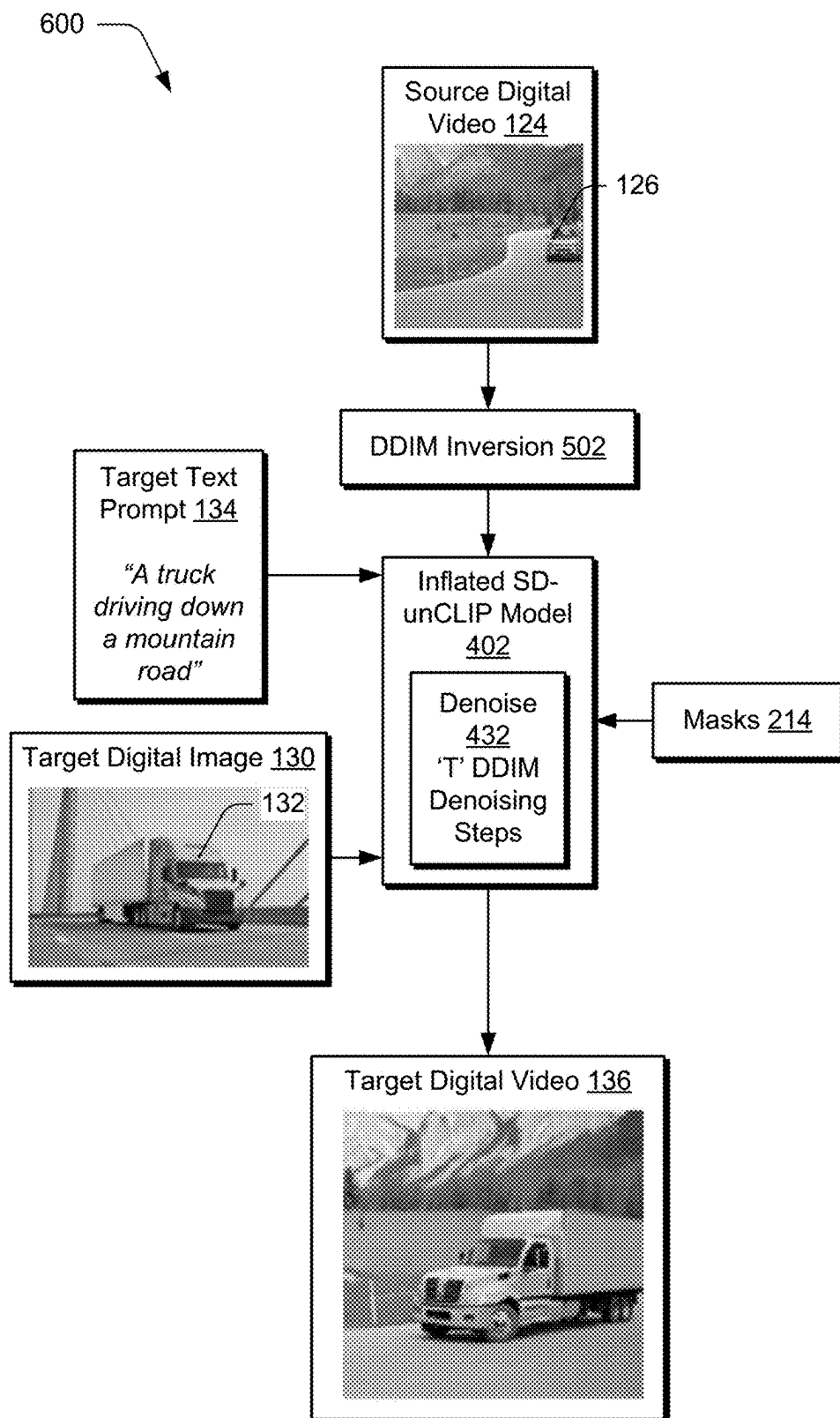


Fig. 6

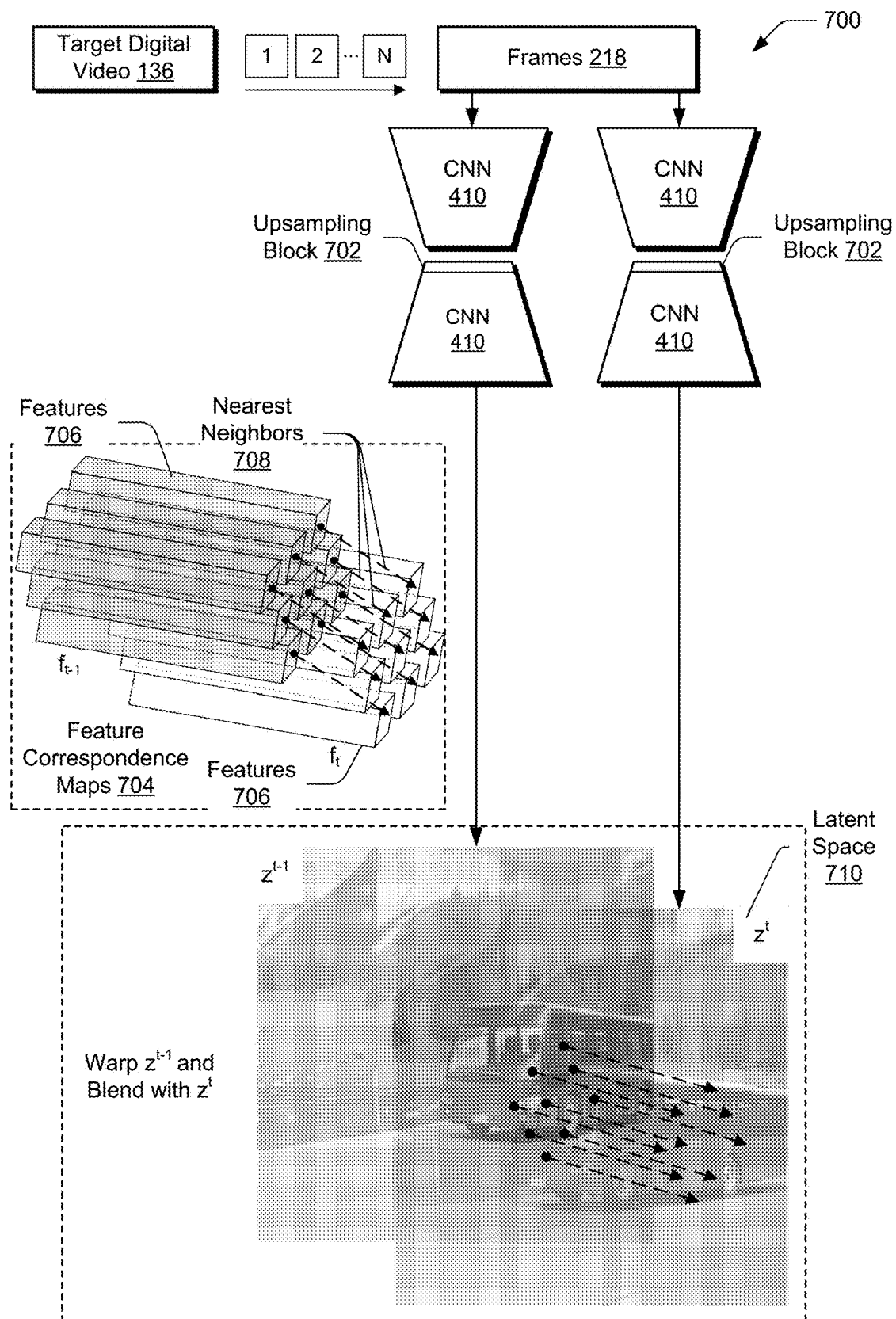


Fig. 7



800

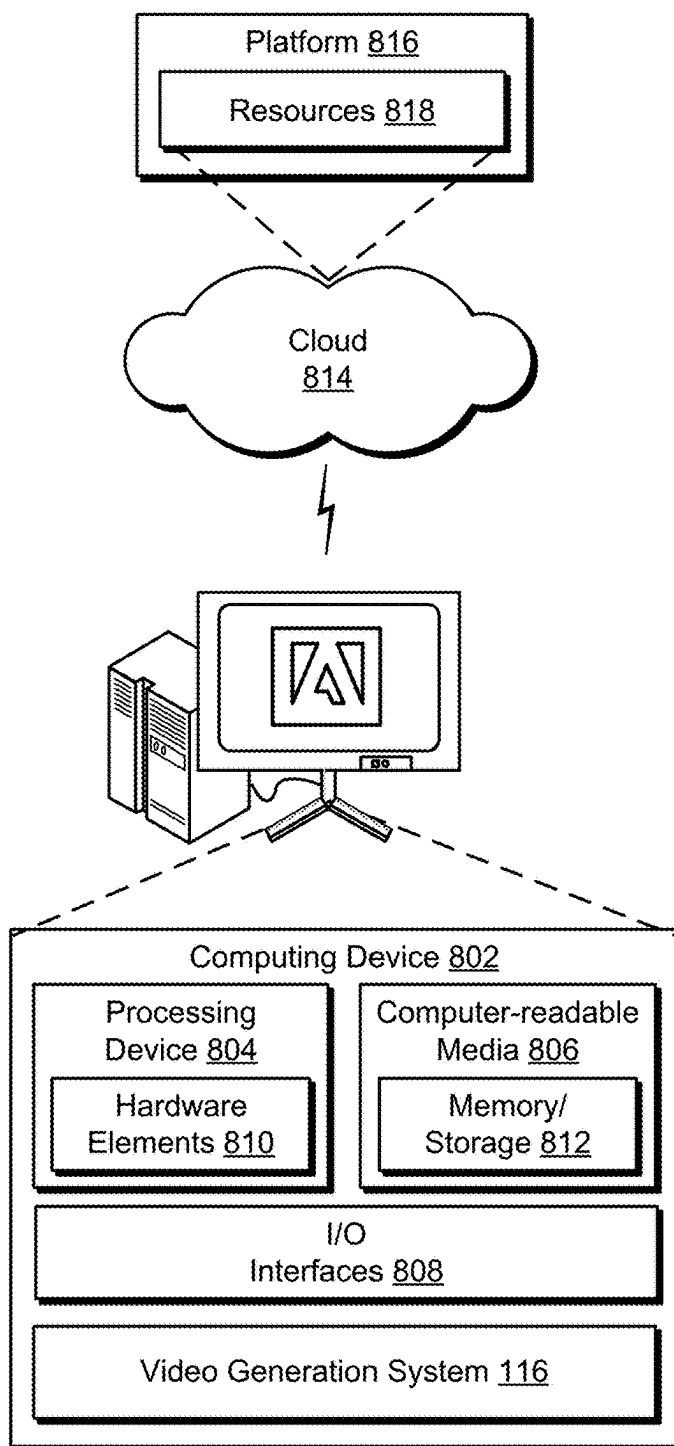


Fig. 8

## DIGITAL VIDEO EDITING BASED ON A TARGET DIGITAL IMAGE

### BACKGROUND

[0001] Conventional techniques that rely on machine-learning models to perform digital video edits are confronted with numerous technical challenges that reduce accuracy in achieving a desired output. Digital video, for instance, introduces additional technical challenges over editing of singular digital images. This is because generation of digital video involves visual and temporal consistency that is to be maintained between consecutive digital images that form frames of the digital video.

[0002] Additionally, conventional digital video editing techniques are limited by an amount of expressive power that is supported in specifying how to perform the edits. Accordingly, conventional techniques are hindered in an ability to define, typically using text, as to “what is to be performed” as part of the edit by a respective machine-learning model. These technical challenges are further aggravated when confronted with edits involving objects having differing sizes and shapes, one to another. Conventional techniques, for instance, often fail in replacement of an object having a shape and size that is different from a shape and size of another object that is to serve as the replacement. Because of this, conventional digital video editing techniques often result in visual inaccuracies, incur computational inefficiencies, and result in increased power consumption.

### SUMMARY

[0003] Digital video editing techniques are described that are based on a target digital image. In one or more implementations, inputs are received. The inputs include a target text prompt, a target digital image depicting a target object, and a source digital video having a plurality of frames depicting a source object. Regions-of-interest are identified in the plurality of frames of the source digital video, respectively, based on the target text prompt and the target digital image using a machine-learning model, e.g., a diffusion model. A plurality of frames of a target digital video are generated as having the target object using a generative machine-learning model. The generating is based on the regions-of-interest, the target digital image, the source digital video, and a source text prompt describing the source digital video.

[0004] This Summary introduces a selection of concepts in a simplified form that are further described below in the Detailed Description. As such, this Summary is not intended to identify essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The detailed description is described with reference to the accompanying figures. Entities represented in the figures are indicative of one or more entities and thus reference is made interchangeably to single or plural forms of the entities in the discussion.

[0006] FIG. 1 is an illustration of a digital medium environment in an example implementation that is operable to employ digital video editing based on a target digital image as described herein.

[0007] FIG. 2 depicts a system in an example implementation showing operation of a video generation system of FIG. 1 in greater detail.

[0008] FIG. 3 is a flow diagram depicting an algorithm as a step-by-step procedure in an example implementation of operations performable for accomplishing a result of target digital video generation based on a source digital video, a source text prompt, a target digital image, and a target text prompt.

[0009] FIG. 4 depicts a system in an example implementation showing training of a machine-learning model of a region-of-interest identification module of FIG. 2.

[0010] FIG. 5 depicts a system in an example implementation showing operation of a generation module of the video generation system of FIG. 2 in greater detail.

[0011] FIG. 6 depicts a system in an example implementation showing operation of a frame generation module of FIG. 2 in greater detail as performing mask guided inference.

[0012] FIG. 7 depicts a system in an example implementation showing operation of a latent correction module of the video generation system of FIG. 2 in greater detail.

[0013] FIG. 8 illustrates an example system including various components of an example device that can be implemented as any type of computing device as described and/or utilize with reference to the previous figures to implement embodiments of the techniques described herein.

### DETAILED DESCRIPTION

#### Overview

[0014] Conventional generative techniques used to support digital video editing are confronted with numerous technical challenges. These technical challenges limit accuracy of digital video editing, do not support temporal consistency between frames of the digital video, and are incapable of addressing object replacement using varying shapes and sizes.

[0015] Conventional techniques that are based on diffusion models, for instance, rely solely on a text prompt for specifying an edit. A diffusion model is a type of generative machine-learning model that is used for digital content creation. In order to train a diffusion model, noise is added to training data samples until the data within the training data samples is obscured. The diffusion model is then trained to reverse this process based on training data that also has a text prompt that describes the digital content to be created in order to generate data samples as the digital content that corresponds to the text prompt.

[0016] Conventional techniques that rely on diffusion models for content creation, therefore, are unsuitable for scenarios in which a nature of the edit cannot be accurately expressed using text, solely. As a result, the limited expressive power of the text prompt relied upon by these conventional techniques limits accuracy in performing an edit. Additionally, conventional techniques lack shape awareness and therefore fail in scenarios in which a source object to be replaced has a size and/or shape is substantially different than a target object that is to replace the source object. Further, conventional techniques fail to enforce inter-frame temporal consistency and thus result in visual artifacts that are readily noticeable to a human being when viewing the digital video.

[0017] Accordingly, digital video editing techniques that are based on a target digital image are described. These techniques are configurable to leverage a target digital image depicting a target object as a visual guide to improve accuracy in editing a source digital video in order to generate a target digital video. In this way, the target object, as depicted in the target digital image, expands expressiveness supported in editing the source digital video. The expanded expressiveness also supports a variety of functionalities that are not supported by conventional digital video editing techniques, including an ability of handle edits involving objects having different size and shapes, an ability to maintain temporal consistency between frames of the target digital video being generated, and so forth. The digital video editing techniques, for instance, are usable to replace a source object in a digital video with a target object having a different size and shape as following movement exhibited by the source object in the source digital video. In this way, the digital video editing techniques overcome technical challenges of conventional techniques as part of digital video generation.

[0018] In one or more examples, inputs are received by a video generation system that is configured to generate a target digital video, e.g., using generative artificial intelligence as implemented using one or more machine-learning models. The inputs include a target text prompt, a target digital image depicting a target object, a source digital video having a plurality of frames depicting a source object, and a source text prompt describing the source digital video.

[0019] The source digital video, for instance, depicts a source object of a white bus as driving down a road having a background of a mountain scene. The source text prompt describes the source digital video as “Driving a white bus down a mountain road.” The target digital image includes a red sport utility vehicle (SUV) and the target text prompt describes “Driving a red SUV down a mountain road.” In this example, the video generation system is tasked with replacing the white bus with the red SUV as expressed by the target text prompt and depicted in the target digital image as part of generating a target digital video.

[0020] The video generation system begins by first identifying regions-of-interest in the frames of the source digital video. To do so in one or more examples, the video generation system employs one or more diffusion models using generative artificial intelligence (AI) techniques to generate masks of the respective frames. The one or more diffusion models, for instance, include a source denoising branch configured to process the source text prompt and a target denoising branching configured to process the target text prompt and the target object of the target digital image.

[0021] The source digital video is then transformed using randomized latent noise, which is then denoised by the respective branches of the one or more diffusion models. Differences in the denoising operations as performed across respective timesteps in the respective branches are compared and then used (e.g., as a reconstructive loss) as a basis to form the masks, e.g., which are averaged and binarized to define the regions-of-interest. In this way, the regions-of-interest are generated by the video generation system as “target aware” by leveraging knowledge provided by the target object in the target digital image, which is not possible in conventional solely text-based techniques. The masks, as identifying the regions-of-interest, are injected into a subsequent machine-learning model (e.g., as embeddings) along

with indications of respective timesteps (e.g., also as embeddings) in order to generate the target digital video.

[0022] Although the masks are configured to accurately identify the regions-of-interest that are to be a subject of an edit, the masks themselves do not address temporal consistency of an object within the regions-of-interest across the generated frames. Continuing with the above example, while a shape of an edit from a “white bus” to a “red SUV” across consecutive frames may appear generally similar, a different stylistic appearance of the shape may be employed to provide visual consistency. Although conventional techniques have been developed that employ optical flow, these conventional techniques exhibit inaccuracies in scenarios involving different shapes or different stylistic appearances.

[0023] Accordingly, the video generation system in this example is configured to implement a latent correction strategy during inference in generation of the target digital video. The video generation system, for example, is also configured to utilize a generative machine-learning model implemented using a diffusion model to generate the target digital video. To do so, the diffusion model takes as an input the regions-of-interest (e.g., the masks as described above), the target digital image, the source digital video, and the source text prompt. The video generation system then performs denoising operations on the regions-of-interest as part of including the target object in respective frames of the target digital video.

[0024] As part of generating the frames of the target digital video, a three-step process is performed as part of a latent correction strategy by the video generation system to promote temporal consistency. First, an inter-frame latent field is computed based on features between consecutive frames of the target digital video, e.g., as nearest neighbors. The inter-frame latent field therefore defines a mapping from spatial locations of features in a first frame to its nearest neighbor (e.g., in terms of cosine similarity) to features in a second frame, e.g., that follows the first frame consecutively in a sequence.

[0025] Second, the video generation system blends the computed inter-frame latent fields of adjacent frames inside the regions-of-interest, i.e., the mask regions. The video generation system, for instance, employs a decoder of a machine-learning model to perform the blending at each inference timestep corresponding to respective frames of the target digital video. Third, the video generation system is configured to preserve a background of the source digital video in one or more examples by limiting the denoising operations to the regions-of-interest defined by the masks.

[0026] In this way, the video generation system is configured to generate a target digital video having a target object that follows movement of a source object within a source digital video. To achieve this functionality, the video generation system is configured to expand expressiveness supported in editing the source digital video through use of a target digital image that depicts the target object. The expanded expressiveness also supports a variety of functionalities that are not supported by conventional digital video editing techniques, including an ability of handle edits involving objects having different size and shapes, maintaining temporal consistency between frames of the target digital video being generated, and so forth. Further discussion of these and other examples is included in the following sections and shown in corresponding figures.

### Term Examples

**[0027]** A “machine-learning model” refers to a computer representation that can be tuned (e.g., trained and retrained) based on inputs to approximate unknown functions. In particular, the term machine-learning model can include a model that utilizes algorithms to learn from, and make predictions on, known data by analyzing training data to learn and relearn to generate outputs that reflect patterns and attributes of the training data. Examples of machine-learning models include neural networks, convolutional neural networks (CNNs), long short-term memory (LSTM) neural networks, decision trees, and so forth.

**[0028]** A “diffusion model” is a type of generative machine-learning model that is used for digital content creation. In order to train a diffusion model, noise is added to training data samples until the data within the training data samples is obscured. The diffusion model is then trained to reverse this process based on training data that also has a text prompt that describes the digital content to be created in order to generate data samples as the digital content that corresponds to the text prompt.

**[0029]** In the following discussion, an example environment is described that employs the techniques described herein. Example procedures are also described that are performable in the example environment as well as other environments. Consequently, performance of the example procedures is not limited to the example environment and the example environment is not limited to performance of the example procedures.

### Example Digital Video Generation Environment

**[0030]** FIG. 1 is an illustration of a digital medium environment 100 in an example implementation that is operable to employ digital video editing based on a target digital image as described herein. The illustrated environment 100 includes a service provider system 102 and a computing device 104 that are communicatively coupled, one to another, via a network 106. Computing devices are configurable in a variety of ways.

**[0031]** A computing device, for instance, is configurable as a desktop computer, a laptop computer, a mobile device (e.g., assuming a handheld configuration such as a tablet or mobile phone), and so forth. Thus, a computing device ranges from full resource devices with substantial memory and processor resources (e.g., personal computers, game consoles) to a low-resource device with limited memory and/or processing resources (e.g., mobile devices). Additionally, although a single computing device is shown and described in instances in the following discussion, a computing device is also representative of a plurality of different devices, such as multiple servers utilized by a business to perform operations “over the cloud” for the service provider system 102 and as further described in relation to FIG. 8.

**[0032]** The service provider system 102 includes a digital service manager module 108 that is implemented using hardware and software resources 110 (e.g., a processing device and computer-readable storage medium) in support one or more digital services 112. Digital services 112 are made available, remotely, via the network 106 to computing devices, e.g., computing device 104. Digital services 112 are scalable through implementation by the hardware and software resources 110 and support a variety of functionalities, including accessibility, verification, real-time processing,

analytics, load balancing, and so forth. Examples of digital services include a social media service, streaming service, digital content repository service, content collaboration service, and so on. Accordingly, in the illustrated example, a communication module 114 (e.g., browser, network-enabled application, and so on) is utilized by the computing device 104 to access the one or more digital services 112 via the network 106. A result of processing using the digital services 112 is then returned to the computing device 104 via the network 106.

**[0033]** In the illustrated example, the digital services 112 are utilized to implement a video generation system 116 that is configured to generate a digital video 118, which is stored in a storage device 120. Although illustrated as implemented remotely by the service provider system 102, functionality of the video generation system 116 is also configurable for implementation locally, e.g., as part of the communication module 114 at the computing device 104. The video generation system 116 is configured to leverage generative artificial intelligence (AI) techniques implemented using a machine-learning model (e.g., one or more diffusion models) to generate the digital video 118.

**[0034]** To do so, the video generation system 116 receives inputs 122, e.g., from the computing device 104. The inputs 122 include a source digital video 124 having a source object 126, a source text prompt 128, a target digital image 130 having a target object 132, and a target text prompt 134. The video generation system 116 then leverages insights and expressiveness supported by the target object 132 in order to generate a target digital video 136. In the target digital video 136, in one or more examples, the target object 132 follows movement of the source object 126 in the source digital video 124, thereby functioning as an edit to the source digital video 124.

**[0035]** As previously described, conventional generative digital video editing techniques that employ diffusion models rely solely on a text prompt. Accordingly, conventional techniques are limited by an expressiveness supported by text. As a result, conventional techniques also struggle to accurately edit a digital video when a size and shape between a source object that is to be replaced and a target object used to replace the source object differ.

**[0036]** Accordingly, to address these and other technical challenges, the video generation system 116 is configured to employ a target digital image 130 depicting a target object 132 as part of generating the target digital video 136. Through use of the target digital image 130, and more particularly identification of the target object 132 within the target digital image 130, the video generation system 116 is configurable to overcome conventional technical challenges through increased expressiveness of the target object 132 over conventional techniques that are limited to text, alone. As a result, the video generation system 116 is configured to overcome conventional technical challenges in support of digital video generation to address variances in shapes and sizes as well as promote temporal consistency between frames of the target digital video 136.

**[0037]** As displayed by a display device 138 in a user interface 140 of the computing device 104, for instance, a source digital video 142 depicts a person’s feet as walking and wearing gray shoes. A target digital image 144 depicts a target object as a red shoe. The source digital video 142 and the target digital image 144 are usable by the video generation system 116 to generate a target digital video 146

of the person's feet that are walking and wearing red shoes. Examples of inputs **122** provided in this example further include a source text prompt that uses text to describe what is depicted in the source digital video, e.g., "gray shoes walking in a park" **148**. The inputs **122** also include a target text prompt describing an edit involving what is to be depicted in the target digital video **146**, e.g., "red shoes worn for a walk in a park" **150**. The target digital video **146** is then generated in this example as replacing a source object (e.g., the gray shoes) with a target object (e.g., the red shoes) as following motion of the source object in the source digital video **142**. The video generation system **116** is able to do so even though shapes of the gray shoes and the red shoe may vary between different frames of the target digital video **146**. Further, the video generation system **116** is also configurable to promote temporal consistency between frames of the target digital video **146**. Further discussion of operation of the video generation system **116** as performing digital video editing based on a target digital image is described in the following section and shown in corresponding figures.

**[0038]** In general, functionality, features, and concepts described in relation to the examples above and below are employed in the context of the example procedures described in this section. Further, functionality, features, and concepts described in relation to different figures and examples in this document are interchangeable among one another and are not limited to implementation in the context of a particular figure or procedure. Moreover, blocks associated with different representative procedures and corresponding figures herein are applicable together and/or combinable in different ways. Thus, individual functionality, features, and concepts described in relation to different example environments, devices, components, figures, and procedures herein are usable in any suitable combinations and are not limited to the particular combinations represented by the enumerated examples in this description.

#### Example Digital Video Generation Guided by a Target Digital Image

**[0039]** The following discussion describes digital video generation techniques that are implementable utilizing the described systems and devices. Aspects of each of the procedures are implemented in hardware, firmware, software, or a combination thereof. The procedures are shown as a set of blocks that specify operations performable by hardware and are not necessarily limited to the orders shown for performing the operations by the respective blocks. Blocks of the procedures, for instance, specify operations programmable by hardware (e.g., processor, microprocessor, controller, firmware) as instructions thereby creating a special purpose machine for carrying out an algorithm as illustrated by the flow diagram. As a result, the instructions are storable on a computer-readable storage medium that, in response to execution by a processing device, causes the hardware to perform the algorithm.

**[0040]** FIG. 2 depicts a system **200** in an example implementation showing operation of a video generation system **116** of FIG. 1 in greater detail. FIG. 3 is a flow diagram depicting an algorithm **300** as a step-by-step procedure in an example implementation of operations performable for accomplishing a result of target digital video generation based on a source digital video, a source text prompt, a target digital image, and a target text prompt. Portions of the

algorithm **300** are described in parallel in the following discussion as part of describing operation of the system **200** of FIG. 2.

**[0041]** Use of diffusion models has gained popularity in scenarios involving editing of static digital images (i.e., a single digital image) using text prompts. Although success has been exhibited in these scenarios, these techniques often fail when confronted with digital video editing tasks as focused exclusively on use of text to describe the edits. Accordingly, as previously described these conventional techniques often fail in scenarios in which a nature of the edit cannot be accurately expressed using text, alone. Further, conventional techniques lack shape awareness and therefore also fail in instances in which a shape and/or size of a target object differs substantially from a shape and/or size of a source object in a source digital video.

**[0042]** The video generation system **116**, therefore, is configurable to address these and other technical challenges. The video generation system **116**, for instance, is configurable to identify regions-of-interest in frames of a source digital video, respectively. To do so, the video generation system **116** is configurable to employ a diffusion model that processes the source digital video based on a target text prompt and the target digital image. In one or more examples, the diffusion model does so after being inflated and trained according to a one-shot finetuning approach as further described below in relation to FIG. 4.

**[0043]** The video generation system **116** is also configurable to address technical challenges involving maintenance of temporal consistency between frames of the target digital video being generated, i.e., the edited digital video. Conventional techniques are guided by the source digital video to maintain temporal consistency, e.g., by using source-based neural-layer atlases, source-based inter-frame feature propagation, and so on. However, these conventional techniques fail in instances in which a target object substantially differs from a source object being replaced, e.g., by size and/or shape.

**[0044]** To address these technical challenges, the video generation system **116** is configurable to implement a latent correction strategy to blend inter-frame latent fields computed by the diffusion model "on the fly" (i.e., in real time) during inference to improve inter-frame temporal consistency of the target object in the target digital video. The video generation system **116**, for instance, is configurable to employ guidance based on identification of the regions-of-interest (e.g., masks) to support temporal consistency even in instances in which a target object has a different shape and/or size than a source object in a source digital video, which is not possible using conventional techniques.

**[0045]** To begin in the example system **200** of FIG. 2, an input module **202** receives a plurality of inputs **122** (block **302**). The inputs **122** includes a source digital video **124**, e.g., " $V^{src}=[I_1^{src}, \dots, I_N^{src}]$ " having " $N$ " frames containing a source object **126**. The inputs **122** also include a source text prompt **128** (e.g., " $P^{src}$ ") that describes the source digital video **124**. A target digital image **130** " $I^{tgt}$ " is also received as part of the inputs **122** and includes a target object **132**. The inputs **122** further include a target text prompt **134** (e.g., " $P^{tgt}$ ") describing an edit to be made to the source digital video **124**. The video generation system **116** generates a target digital video **136** " $V^{tgt}=[I_1^{tgt}, \dots, I_N^{tgt}]$ " which preserves motion of the source digital video **124** but replaces

the source object **126** with a target object **132** from the target digital image **130** in this example.

[0046] The inputs **122** are then passed to a region-of-interest identification module **204**. The region-of-interest identification module **204** is configured to identify regions-of-interest **206** in a plurality of frames, respectively of the source digital video **124** based on the target text prompt **134** and the target digital image **130** using a machine-learning model **208**. The machine-learning model **208**, for example is configurable as a diffusion model **210**, which is trainable in a variety of ways, an example of which is described as follows and shown in a corresponding figure.

[0047] FIG. 4 depicts a system **400** in an example implementation showing training of the machine-learning model **208** of the region-of-interest identification module **204** of FIG. 2. In this example, the machine-learning model **208** is implemented as a stable diffusion text-to-image model, shown as the inflated SD-unCLIP model **402**. The inflated SD-unCLIP model **402** accepts CLIP image embeddings and CLIP text embedding as conditional inputs. The inflated SD-unCLIP model **402** conditions video generation on the source text prompt **128** and a reference digital image **304** taken from the source digital video **124**. The source text prompt **128** provides context to the inflated SD-unCLIP model **402** about the source digital video **124**, such as “a person wearing gray shoes,” “a women wearing gray shoes,” and the like. The reference digital image **404** is sampled as a frame (e.g., randomly) from the source digital video **124**. In the illustrated example, the inflated SD-unCLIP model **402** implements a CLIP text conditional model (illustrated as “CLIP text **406**”) for receiving the source text prompt **128** and a CLIP image conditional model (illustrated as “CLIP image **408**”) for receiving the reference digital image **404**. [0048] The inflated SD-unCLIP model **402** is implemented, at least in part, using a convolutional neural network **410** in this example. In some implementations, the convolutional neural network **410** is or uses an architecture similar to a U-Net. The convolutional neural network **410**, for example, is configurable to use an architecture that follows an encoder-decoder structure in which a contracting path represents an encoder **412** and an expanding path represents a decoder **414**.

[0049] The convolutional neural network **410** receives the source digital video **124** having noise **434** added, the source text prompt **128** via the CLIP text **406** conditional model, and the reference digital image **404** via the CLIP image **408** conditional model. The convolutional neural network **410** processes this data through various components, shown as convolutional blocks **416** (in gray) and attention blocks **418** (in white). The convolutional blocks **416** process and extract features from the input data by applying filters and reducing data dimensions to achieve efficient and effective feature representation. The attention blocks **418** enhance the ability of the convolutional neural network **410** to selectively focus on and emphasize relevant features in a given input.

[0050] In the illustrated example, the attention blocks **418** include a spatio-temporal attention block (ST-Attn block **420**) that enables the convolutional neural network **410** to focus on specific spatial (i.e., location-based) and temporal (i.e., time-based) aspects of the input data frames simultaneously. A cross attention block (C-Attn block **422**) enables the convolutional neural network **410** to process multiple data types, such as text and image data. Processing of the multiple data types allows the inflated SD-unCLIP model

**402** to focus on relevant parts of one input (e.g., the source text prompt **128**) based on information from another input, e.g., the reference digital image **404**. In this manner, the inflated SD-unCLIP model **402** effectively integrates and processes data across different data types.

[0051] An additional temporal attention block (T-Attn block **424**) is introduced after the ST-Attn block **420** and the C-Attn block **422**, i.e., a cross attention block. The T-Attn block **424** enhances the ability of the convolutional neural network **410** to process and interpret sequential or time-series data, such as the video frames included as part of the source digital video **124**. In one or more implementations, parameters of the ST-Attn block **420**, the C-Attn block **422**, and the T-Attn block **424** are fine-tuned while keeping the convolutional blocks **416** frozen.

[0052] In particular, query weights (illustrated as Q **426**) for the ST-Attn block **420** and the C-Attn block **422** are updated while key/value weights (illustrated as K/V **428**) are left unchanged. In addition, each of the query weights (Q **426**) and key/value weights (K/V **428**) for the T-Attn block **424** are updated. Updated weights for each type of attention block **418** are depicted in gray in FIG. 4.

[0053] The query weights (Q **426**) are parameters in the convolutional neural network **410** that transform input data into a query representation and used to determine the relevance of different parts of the input. The key/value weights (K/V **428**) are parameters in the convolutional neural network **410** that transform input data into “key” and “value” representations, where key representations are used to match with queries and value representations carry the information to be focused on after matching.

[0054] Fine-tuning the query weights (Q **426**) and the key/value weights (K/V **428**) enables optimization of the inflated SD-unCLIP model **402** for target-aware and temporally consistent generation of the target digital video **136**. Specifically, fine-tuning of the weights allows the inflated SD-unCLIP model **402** to better adapt to the nuances of dynamics of the source digital video **124**, enhancing the ability of the inflated SD-unCLIP model **402** to accurately weigh the importance of different parts of the inputs **122** and to make precise predictions.

[0055] The illustrated example shows the convolutional neural network **410** configured as a feed forward network, depicted as “FFN **430**.” A feed forward network is type of neural network architecture where connections between nodes (i.e., neurons) move in a single direction, from input to output, without forming a loop. In the example neural network architecture, the feed forward network (FFN **430**) is augmented with the convolutional blocks **416** and the attention blocks **418**. The convolutional blocks **416**, which are adept at processing visual information, act as feature extractors that capture spatial hierarchies in data, such as edges or textures in the reference digital image **404**. These extracted features are then passed through the feedforward layers for further processing.

[0056] The attention blocks **418**, which are effective in sequence processing tasks like text or time series analysis and spatial processing tasks, enable the convolutional neural network **410** to focus selectively on different parts of the input. By integrating these blocks, the feed forward network (FFN **430**) combines the spatial processing capabilities of a convolutional neural network and the sequence-focused processing of attention mechanisms, enhancing the ability of

the inflated SD-unCLIP model **402** to handle complex tasks that involve both feature extraction and focused attention on specific input segments.

**[0057]** The convolutional neural network **410**, once trained as an example of the machine-learning model **208** of FIG. 2, is then employed to generate the regions-of-interest **206** from the source digital video **124** based on the target object **132** of the target digital image **130** and the target text prompt **134**. The region-of-interest identification module **204**, for instance, employs a mask generation module **212** that is configured to employ the machine-learning model **208** to form a plurality of masks **214** defining, respectively, the regions-of-interest **206** (block **306**). To do so in one or more examples, noise differences are compared as a reconstruction loss across respective timesteps between a source denoising branch and a target denoising branch of one or more diffusion models **210** (block **308**). Further discussion of use of a noise reconstruction loss is included in the following description and shown in a corresponding figure.

**[0058]** FIG. 5 depicts a system **500** in an example implementation showing operation of the mask generation module **212** of the video generation system **116** of FIG. 2 in greater detail. A source denoising branch of the diffusion model **210** of FIG. 2 is illustrated as receiving a source text prompt **128** and a source digital video **124** having a source object **126**. A target denoising branch, on the other hand, is illustrated as receiving a target text prompt **134** and a target digital image **130** depicting a target object **132**.

**[0059]** The system **500** in the illustrated example uses a fine-tuned version of the inflated SD-unCLIP model **402** as described previously in relation to FIG. 4. The inflated SD-unCLIP model **402** applies a denoising diffusion implicit model inversion process (illustrated as “DDIM inversion **502**”) to produce noisy latents **504** from frames of the source digital video **124**. The noisy latents **504** are latent representations of the frames of the source digital video **124** after being intentionally altered by adding noise **434**, e.g., Gaussian noise. In one or more examples, the inflated SD-unCLIP model **402** systematically introduces the noise **434** to the frames of the source digital video **124**, creating progressively noisier versions until a state of solely noise is reached.

**[0060]** The system **500**, in one or more examples, performs a denoise process (illustrated as “denoise **432**” in FIG. 4) of the noisy latents **504** under different conditions using deterministic DDIM sampling. Initially, the denoising **432** is guided by the source text prompt **128** and a frame of the source digital video **124** as the reference digital image **404** input to the inflated SD-unCLIP model **402** via the CLIP text **406** conditional model and the CLIP image **408** conditional model, respectively. Subsequently, the denoise **432** process is repeated using the target text prompt **134** and the target digital image **130** that includes the target object **132** as conditional inputs to the inflated SD-unCLIP model **402** via the CLIP text **406** conditional model and the CLIP image **408** conditional model, respectively.

**[0061]** The system **500** then computes differences (illustrated individually as compute difference **506**) in the noise predicted by the convolutional neural network **410** at each denoising timestep of these two separate DDIM samplings from the respective branches. The computed differences are depicted as heat maps **508** in the illustrated example. The computed differences are then averaged over the denoising time steps and binarized (illustrated as average differences/binarize **510**) to generate the masks **214** for each frame of

the source digital video **124**. The masks **214**, once generated indicate the regions-of-interest, to which, the edits are to be applied. For example, the masks **214** are configured to indicate that edits are to be applied to an area resembling a truck rather than a car, recognizing the truck’s larger size in comparison to the car as well as a different shape in comparison with the car.

**[0062]** Returning again to FIG. 2, the regions-of-interest **206** (e.g., the masks **214**) are then passed by the region-of-interest identification module **204** to a frame generation module **216** to generate a plurality of frames **218** of a target digital video **136**, e.g., as forming an edited version of the source digital video **124**. To do so, the frame generation module **216** employs a generative machine-learning model **220**, illustrated examples of which include one or more diffusion models (block **310**). The generating is based on the regions-of-interest **206**, the target digital image **130**, the source digital video **124**, and a source text prompt **128** describing the source digital video **124**.

**[0063]** FIG. 6 depicts a system **600** in an example implementation showing operation of the frame generation module **216** of FIG. 2 in greater detail as performing mask guided inference using the inflated SD-unCLIP model **402** as implementing the one or more diffusion models **222** of the generative machine-learning model **220**. The inflated SD-unCLIP model **402** receives, as input, the source digital video **124** after being processed using DDIM inversion **502**. The inflated SD-unCLIP model **402** also receives the target text prompt **134**, the target object **132** as depicted by the target digital image **130**, and the masks **214**.

**[0064]** In the illustrated example, the source digital video **124** contains video frames of a source object **126** (e.g., a car) driving down a mountain road. The target text prompt **134** specifies that the inflated SD-unCLIP model **402** is to generate the target digital video **136** as an edited version of the source digital video **124** of “a truck driving down a mountain road.” The target digital image **130** is an image of a truck to replace the source object **26** of the car in the source digital video **124**.

**[0065]** The masks **214** are used to isolate features of the car to be replaced by features of the truck while maintaining the correct pose, shape, and style in the target digital video **136**. Prior to generating the target digital video **136**, an initial output of the inflated SD-unCLIP model **402** is denoised **432** in “T” DDIM denoising steps. In particular, the denoise **432** process is carried out “T” times (i.e., “T” denoising steps) using the inflated SD-unCLIP model **402**. The same inflated SD-unCLIP model **402** is used during each denoising step in this example. The input “t” differs at each denoising step “t.” The output of the “t” denoising step is fed as input to the “t+1” denoising step. A result of which is a target digital video **136** depicted as having the target object **132** which replaces the source object **126** in the source digital video **124**.

**[0066]** Although the masks **214** are configured to accurately identify the regions-of-interest **206** that are to be a subject of an edit, the masks **214** themselves do not address temporal consistency of the target object **132** within the regions-of-interest **206** across the generated frames **218**. In the illustrated example, while a shape of an edit from a “truck” to a “car” across consecutive frames may appear generally similar, a different stylistic appearance of the shape may be employed to provide visual consistency. Although conventional techniques have been developed that

employ optical flow, these conventional techniques exhibit inaccuracies in scenarios involving different shapes or different stylistic appearances.

[0067] Returning again to FIG. 2, the frame generation module 216 in this example is configured to implement a latent correction strategy during inference in generation of the target digital video, functionality of which is represented as a latent correction module 224. As part of generating the frames 218 of the target digital video 136, a three-step process is performed as part of the latent correction strategy to promote temporal consistency (block 312). First, an inter-frame latent field is calculated by an inter-field computation module 226 based on features between consecutive frames 218 of the target digital video 136 (block 314), e.g., as nearest neighbors. The inter-frame latent field therefore defines a mapping from spatial locations of features in a first frame to its nearest neighbor (e.g., in terms of cosine similarity) to features in a second frame, e.g., that follows the first frame consecutively in a sequence.

[0068] Second, the frame generation module 216 utilizes a blending module to blend the computed inter-frame latent fields of adjacent frames inside the regions-of-interest 206 (block 316), i.e., as defined by the masks 214. The blending module 228, for instance, employs a decoder of a machine-learning model to perform the blending at each inference timestep corresponding to respective frames of the target digital video.

[0069] Third, a background preservation module 230 is leveraged to preserve a background of the source digital video 124 by limiting the denoising operations to the regions-of-interest defined by the masks 214. The target digital video 136 is then output (block 318) as having the latent correction applied to the plurality of frames 218. Further discussion of applying a latent correction is included in the following description and shown in a corresponding figure.

[0070] FIG. 7 depicts a system 700 in an example implementation showing operation of the latent correction module 224 of the video generation system 116 of FIG. 2 in greater detail. During inference, the latent correction module 224 implements a feature blending strategy in the latent space of the inflated SD-unCLIP model 402 to improve inter-frame temporal consistency of the target object 132 in the target digital video 136. This is a process in which a feature correspondence map is computed and then features are blended using the feature correspondence map.

[0071] During each denoising timestep “t” during inference, the latent correction module 224 utilizes the features of an upsampling block 702 of the convolutional neural network 410 for estimating feature correspondence maps 704 between consecutive frames 218 of the target digital video 136. The upsampling block 702 is implemented, for example, as part of the convolutional blocks 416 of the decoder 414 portion of the convolutional neural network 410.

[0072] For the target digital video 136 with “N” frames, “[f<sub>1</sub><sup>t</sup>, . . . f<sub>N</sub><sup>t</sup>]” are features 706 given by the upsampling block 702 at the denoising timestep “t.” Nearest neighbors 708 (illustrated using dashed lines) are defined by a nearest neighbor’s field  $\mathcal{N}_{\pm}(\bullet)$  as described in the following equation:

$$\mathcal{N}_{\pm}^t[p] := \operatorname{argmin}_q \mathcal{D}(\mathcal{F}_t^t[p], f_{\pm 1}^t[q])$$

[0073] This equation represents a mapping of spatial locations “p” in the features of the “i<sup>th</sup>” frame to the spatial location “q” of its nearest neighbor in the features of the “(i±1)<sup>th</sup>” frame.

[0074] The masks 214 “[M<sub>1</sub>, M<sub>2</sub>, . . . , M<sub>N</sub>]” effectively predict a coarse region in each frame where a target foreground (i.e., the target object 132) is to appear in the target digital video 136. At each timestep “t,” the blending module 228 of the latent correction module 224 blends the features 706 of the consecutive frames 218 of the target digital video 136 in the masked regions in a latent space 710 of the decoder 414 of the convolutional neural network 410. The latent space 710 is denoted as “z<sub>t</sub>” space at timestep “t.” The blended features “z<sub>t</sub>” are given in the equation below,

$$z_t^t[p] =$$

$$w_{-1}(M_t \cdot z_{t-1}^t[\mathcal{N}_{t-}^t(p)]) + w_0(M_t \cdot z_t^t) + w_1(M_t \cdot z_{t+1}^t[\mathcal{N}_{t+}^t(p)]) + (1 - M_t) \cdot z_t^t$$

where “w<sub>-1</sub>”, “w<sub>0</sub>”, “w<sub>+1</sub>” represent weights, respectively, of non-negative hyperparameters that add up to “1” and “ $\mathcal{N}_{\pm}^t[p]$ ” is the nearest neighbors 708 from the above equation that are upsampled to match a dimension of the “z<sub>t</sub>” space. This blending occurs at each timestep “t,” thereby ensuring the target digital video 136 exhibits temporal consistency of the target object 132 between frames 218 of the target digital video 136.

[0075] In this way, the video generation system is configured to generate a target digital video having a target object that follows movement of a source object within a source digital video. To achieve this functionality, the video generation system is configured to expand expressiveness supported in editing the source digital video through use of a target digital image that depicts the target object. The expanded expressiveness also supports a variety of functionalities that are not supported by conventional digital video editing techniques, including an ability to handle edits involving objects having different size and shapes, maintaining temporal consistency between frames of the target digital video being generated, and so forth.

#### Example System and Device

[0076] FIG. 8 illustrates an example system generally at 800 that includes an example computing device 802 that is representative of one or more computing systems and/or devices that implement the various techniques described herein. This is illustrated through inclusion of the video generation system 116. The computing device 802 is configurable, for example, as a server of a service provider, a device associated with a client (e.g., a client device), an on-chip system, and/or any other suitable computing device or computing system.

[0077] The example computing device 802 as illustrated includes a processing device 804, one or more computer-readable media 806, and one or more I/O interface 808 that are communicatively coupled, one to another. Although not shown, the computing device 802 further includes a system bus or other data and command transfer system that couples the various components, one to another. A system bus can include any one or combination of different bus structures, such as a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus



that utilizes any of a variety of bus architectures. A variety of other examples are also contemplated, such as control and data lines.

**[0078]** The processing device **804** is representative of functionality to perform one or more operations using hardware. Accordingly, the processing device **804** is illustrated as including hardware element **810** that is configurable as processors, functional blocks, and so forth. This includes implementation in hardware as an application specific integrated circuit or other logic device formed using one or more semiconductors. The hardware elements **810** are not limited by the materials from which they are formed or the processing mechanisms employed therein. For example, processors are configurable as semiconductor(s) and/or transistors (e.g., electronic integrated circuits (ICs)). In such a context, processor-executable instructions are electronically-executable instructions.

**[0079]** The computer-readable storage media **806** is illustrated as including memory/storage **812** that stores instructions that are executable to cause the processing device **804** to perform operations. The memory/storage **812** represents memory/storage capacity associated with one or more computer-readable media. The memory/storage **812** includes volatile media (such as random access memory (RAM)) and/or nonvolatile media (such as read only memory (ROM), Flash memory, optical disks, magnetic disks, and so forth). The memory/storage **812** includes fixed media (e.g., RAM, ROM, a fixed hard drive, and so on) as well as removable media (e.g., Flash memory, a removable hard drive, an optical disc, and so forth). The computer-readable media **806** is configurable in a variety of other ways as further described below.

**[0080]** Input/output interface(s) **808** are representative of functionality to allow a user to enter commands and information to computing device **802**, and also allow information to be presented to the user and/or other components or devices using various input/output devices. Examples of input devices include a keyboard, a cursor control device (e.g., a mouse), a microphone, a scanner, touch functionality (e.g., capacitive or other sensors that are configured to detect physical touch), a camera (e.g., employing visible or non-visible wavelengths such as infrared frequencies to recognize movement as gestures that do not involve touch), and so forth. Examples of output devices include a display device (e.g., a monitor or projector), speakers, a printer, a network card, tactile-response device, and so forth. Thus, the computing device **802** is configurable in a variety of ways as further described below to support user interaction.

**[0081]** Various techniques are described herein in the general context of software, hardware elements, or program modules. Generally, such modules include routines, programs, objects, elements, components, data structures, and so forth that perform particular tasks or implement particular abstract data types. The terms “module,” “functionality,” and “component” as used herein generally represent software, firmware, hardware, or a combination thereof. The features of the techniques described herein are platform-independent, meaning that the techniques are configurable on a variety of commercial computing platforms having a variety of processors.

**[0082]** An implementation of the described modules and techniques is stored on or transmitted across some form of computer-readable media. The computer-readable media includes a variety of media that is accessed by the computing

device **802**. By way of example, and not limitation, computer-readable media includes “computer-readable storage media” and “computer-readable signal media.”

**[0083]** “Computer-readable storage media” refers to media and/or devices that enable persistent and/or non-transitory storage of information (e.g., instructions are stored thereon that are executable by a processing device) in contrast to mere signal transmission, carrier waves, or signals per se. Thus, computer-readable storage media refers to non-signal bearing media. The computer-readable storage media includes hardware such as volatile and non-volatile, removable and non-removable media and/or storage devices implemented in a method or technology suitable for storage of information such as computer readable instructions, data structures, program modules, logic elements/circuits, or other data. Examples of computer-readable storage media include but are not limited to RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, hard disks, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other storage device, tangible media, or article of manufacture suitable to store the desired information and are accessible by a computer.

**[0084]** “Computer-readable signal media” refers to a signal-bearing medium that is configured to transmit instructions to the hardware of the computing device **802**, such as via a network. Signal media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier waves, data signals, or other transport mechanism. Signal media also include any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media.

**[0085]** As previously described, hardware elements **810** and computer-readable media **806** are representative of modules, programmable device logic and/or fixed device logic implemented in a hardware form that are employed in some embodiments to implement at least some aspects of the techniques described herein, such as to perform one or more instructions. Hardware includes components of an integrated circuit or on-chip system, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a complex programmable logic device (CPLD), and other implementations in silicon or other hardware. In this context, hardware operates as a processing device that performs program tasks defined by instructions and/or logic embodied by the hardware as well as a hardware utilized to store instructions for execution, e.g., the computer-readable storage media described previously.

**[0086]** Combinations of the foregoing are also employed to implement various techniques described herein. Accordingly, software, hardware, or executable modules are implemented as one or more instructions and/or logic embodied on some form of computer-readable storage media and/or by one or more hardware elements **810**. The computing device **802** is configured to implement particular instructions and/or functions corresponding to the software and/or hardware modules. Accordingly, implementation of a module that is executable by the computing device **802** as

software is achieved at least partially in hardware, e.g., through use of computer-readable storage media and/or hardware elements **810** of the processing device **804**. The instructions and/or functions are executable/operable by one or more articles of manufacture (for example, one or more computing devices **802** and/or processing devices **804**) to implement techniques, modules, and examples described herein.

**[0087]** The techniques described herein are supported by various configurations of the computing device **802** and are not limited to the specific examples of the techniques described herein. This functionality is also implementable all or in part through use of a distributed system, such as over a “cloud” **814** via a platform **816** as described below.

**[0088]** The cloud **814** includes and/or is representative of a platform **816** for resources **818**. The platform **816** abstracts underlying functionality of hardware (e.g., servers) and software resources of the cloud **814**. The resources **818** include applications and/or data that can be utilized while computer processing is executed on servers that are remote from the computing device **802**. Resources **818** can also include services provided over the Internet and/or through a subscriber network, such as a cellular or Wi-Fi network.

**[0089]** The platform **816** abstracts resources and functions to connect the computing device **802** with other computing devices. The platform **816** also serves to abstract scaling of resources to provide a corresponding level of scale to encountered demand for the resources **818** that are implemented via the platform **816**. Accordingly, in an interconnected device embodiment, implementation of functionality described herein is distributable throughout the system **800**. For example, the functionality is implementable in part on the computing device **802** as well as via the platform **816** that abstracts the functionality of the cloud **814**.

**[0090]** In implementations, the platform **816** employs a “machine-learning model” that is configured to implement the techniques described herein. A machine-learning model refers to a computer representation that can be tuned (e.g., trained and retrained) based on inputs to approximate unknown functions. In particular, the term machine-learning model can include a model that utilizes algorithms to learn from, and make predictions on, known data by analyzing training data to learn and relearn to generate outputs that reflect patterns and attributes of the training data. Examples of machine-learning models include neural networks, convolutional neural networks (CNNs), long short-term memory (LSTM) neural networks, decision trees, and so forth.

**[0091]** Although the invention has been described in language specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as example forms of implementing the claimed invention.

What is claimed is:

1. A method comprising:

receiving, by a processing device, a target text prompt, a target digital image depicting a target object, and a source digital video having a plurality of frames depicting a source object;

identifying, by the processing device, regions-of-interest in the plurality of frames of the source digital video,

respectively, based on the target text prompt and the target digital image using a machine-learning model;

generating, by the processing device, a plurality of frames of a target digital video having the target object using a generative machine-learning model, the generating based on the regions-of-interest, the target digital image, the source digital video, and a source text prompt describing the source digital video; and

outputting, by the processing device, the target digital video.

2. The method as described in claim 1, wherein the plurality of frames of the target digital video depicts the target object as following motion exhibited by the source object in the source digital video.

3. The method as described in claim 1, wherein the identifying the regions-of-interest includes forming a plurality of masks defining, respectively, the regions-of-interest.

4. The method as described in claim 3, wherein the forming the plurality of masks is based, at least in part, on the target text prompt and the target digital image.

5. The method as described in claim 1, wherein the machine-learning model, utilized to perform the identifying of the regions-of-interest, is configured as one or more diffusion models.

6. The method as described in claim 5, wherein the one or more diffusion models include:

a source denoising branch configured to process the source text prompt; and

a target denoising branch configured to process the target text prompt and the target object of the target digital image.

7. The method as described in claim 6, wherein the identifying includes comparing noise differences as a reconstruction loss across respective timesteps between the source denoising branch and the target denoising branch.

8. The method as described in claim 7, wherein the identifying further comprises averaging and binarizing the noise differences to form a plurality of masks defining, respectively, the regions-of-interest.

9. The method as described in claim 1, wherein the generative machine-learning model, utilized to generate the plurality of frames, is configured as one or more diffusion models.

10. The method as described in claim 1, wherein the generating of the plurality of frames of the target digital video includes calculating a latent correction during inference involving inter-frame temporal consistency.

11. The method as described in claim 10, wherein the calculating includes computing inter-frame latent fields by mapping spatial locations of features between the plurality of frames of the target digital video.

12. The method as described in claim 11, further comprising blending the computed inter-frame latent fields at a plurality of timesteps corresponding to the plurality of frames of the target digital video.

13. The method as described in claim 1, wherein the generating of the plurality of frames of the target digital video includes preserving a background of the source digital video by correcting latent noise corresponding to the background based on the regions-of-interest.

- 14.** A computing device comprising:  
 a processing device; and  
 a computer-readable storage medium storing instructions that, in response to execution by the processing device, causes the processing device to perform operations including:  
 receiving a target text prompt, a target digital image depicting a target object, a source digital video having a plurality of frames depicting a source object, and a source text prompt describing the source digital video;  
 generating a plurality of masks defining regions-of-interest in the plurality of frames of the source digital video using a machine-learning model, the generating based on the source digital video, the target object, the target text prompt, and the source text prompt; and  
 generating a plurality of frames of a target digital video having the target object as following motion of the source object using a generative machine-learning model based on the plurality of masks.
- 15.** The computing device as described in claim **14**, wherein the machine-learning model utilized to perform the generating of the plurality of masks is configured as one or more diffusion models.
- 16.** The computing device as described in claim **15**, wherein the generating of the plurality of masks includes comparing noise differences across respective timesteps between:  
 a source denoising branch of the one or more diffusion models configured to process the source text prompt and frames from the source digital video; and  
 a target denoising branch of the one or more diffusion models configured to process the target text prompt and the target object of the target digital image.
- 17.** The computing device as described in claim **14**, wherein the generating the plurality of frames of the target digital video is performed using a generative machine-

learning model based on the regions-of-interest, the target digital image, the source digital video, and the source text prompt describing the source digital video.

- 18.** One or more computer-readable storage media storing instructions that, in response to execution by a processing device, causes the processing device to perform operations comprising:

receiving a target text prompt, a target digital image depicting a target object, a source digital video having a plurality of frames depicting a source object, and a source text prompt describing the source digital video;  
 generating a plurality of masks defining regions-of-interest in the plurality of frames of the source digital video; and  
 generating a plurality of frames of a target digital video having the target object using a generative machine-learning model, the generating based on the regions-of-interest, the target digital image, the source digital video, and a source text prompt describing the source digital video.

- 19.** The one or more computer-readable storage media as described in claim **18**, wherein the generating a plurality of masks is performed using one or more diffusion models by comparing noise differences across respective timesteps between:

a source denoising branch of the one or more diffusion models configured to process the source text prompt and frames from the source digital video; and  
 a target denoising branch of the one or more diffusion models configured to process the target text prompt and the target object of the target digital image.

- 20.** The one or more computer-readable storage media as described in claim **18**, wherein the generative machine-learning model is configured as a diffusion model.

\* \* \* \* \*