(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 May 2024 (23.05.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/107884 A1

(51) International Patent Classification:
G06T 11/00 (2006.01)

(21) International Application Number:
PCT/US2023/079884

(22) International Filing Date:
15 November 2023 (15.11.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/426,007 16 November 2022 (16.11.2022) US

(71) Applicant: **GOOGLE LLC** [US/US]; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(72) Inventors: **ABERMAN, Kfir**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **PRITCH KNAAN, Yael**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **HERTZ, Amir**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **COHEN-OR, Daniel**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **MOKADY, Ron**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

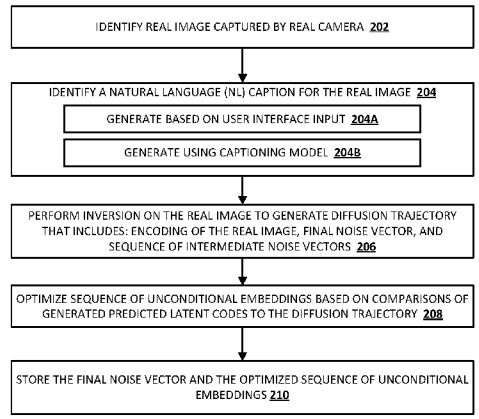
(74) Agent: **HIGDON, Scott** et al.; 3939 Shelbyville Road 201, Louisville, Kentucky 40207 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: NULL-TEXT INVERSION FOR EDITING REAL IMAGES USING GUIDED DIFFUSION MODELS

FIG. 2



202 IDENTIFIER UNE IMAGE RÉELLE CAPTURÉE PAR UNE CAMÉRA RÉELLE
204 IDENTIFIER UN SOUS-TITRE EN LANGAGE NATUREL (NL) POUR L'IMAGE RÉELLE
204A GÉNÉRER SUR LA BASE D'UNE ENTRÉE D'INTERFACE UTILISATEUR
204B GÉNÉRER À L'AIDE D'UN MODÈLE DE SOUS-TITRAGE
206 APPLIQUER UNE INVERSION À L'IMAGE RÉELLE POUR GÉNÉRER UNE TRAJECTOIRE DE DIFFUSION QUI COMPREND : LE CODAGE DE L'IMAGE RÉELLE, DU VECTEUR DE BRUIT FINAL ET DE LA SÉQUENCE DE VECTEURS DE BRUIT INTERMÉDIAIRES
208 OPTIMISER UNE SÉQUENCE D'INTÉGRATIONS NON CONDITIONNELLES SUR LA BASE DE COMPARAISONS DE CODES LATENTS PRÉDITS GÉNÉRÉS À LA TRAJECTOIRE DE DIFFUSION
210 MÉMORISER LE VECTEUR DE BRUIT FINAL ET LA SÉQUENCE OPTIMISÉE D'INTÉGRATIONS NON CONDITIONNELLES

(57) Abstract: Implementations are directed to generating an edited synthetic image, that corresponds to a real image captured using a real camera, but that is generated based on, and reflects, edit(s) to an original natural language (NL) caption for the real image. For example, the original NL caption can be used in performing an inversion on the real image to generate a diffusion trajectory for the real image. Further, the diffusion trajectory can be used to optimize a sequence of unconditional embeddings, for the real image, that are not based on the NL caption for the real image. Yet further, the edited NL caption, the unconditional embeddings, and at least part of a noise vector (of the diffusion trajectory) can be processed, using a Large-scale language-image (LLI) model, to generate the edited synthetic image.

WO 2024/107884 A1

[Continued on next page]

Declarations under Rule 4.17:

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

NULL-TEXT INVERSION FOR EDITING REAL IMAGES USING GUIDED DIFFUSION MODELS

Background

[0001] Large-scale language-image (LLI) models (*e.g.*, text-guided diffusion models), such as GOOGLE'S IMAGEN, have shown phenomenal generative semantic and compositional power, and have gained unprecedented attention from the research community and the public eye. These LLI models are trained on extremely large language-image datasets and use state-of-the-art image generative models, such as auto-regressive and/or diffusion models. These LLI models enable the generation of images conditioned on plain text, known as text-to-image synthesis. For example, these LLI models enable, in response to a plain text prompt of "photo of dog riding on a bicycle", generation of a realistic image that reflects a dog riding on a bicycle. Various LLI models have recently emerged that demonstrate unprecedented semantic generation.

[0002] Image editing is one of the most fundamental tasks in computer graphics, encompassing the process of modifying an input image through the use of an auxiliary input, such as a label, scribble, mask, or reference image.

[0003] However, many LLI models do not provide simple editing means for a generated image, and generally lack control over specific semantic regions of a given image (*e.g.*, using text guidance only). For example, even the slightest change in the textual prompt may lead to a completely different output image being generated using an LLI model. For instance, changing "photo of dog riding on a bicycle" to "photo of white dog riding on a bicycle" can result in a completely different generated image, such as one that changes the dog's shape.

[0004] To circumvent this, many proposed LLI-based editing methods require the user to explicitly mask a part of the image to be inpainted, and drive the edited image to change in the masked area only, while matching the background of the original image. However, the masking procedure is cumbersome (*e.g.*, requiring a large quantity of user inputs to define the mask), hampering quick and intuitive text-driven editing. Moreover, masking the image content removes important structural information, which is completely ignored in the inpainting process. Therefore, some editing capabilities are out of the inpainting scope, such as modifying the texture of a specific object.

[0005] A specifically intuitive way to edit an image is through textual prompt(s) provided by the user. However, some proposed LLI-based editing methods can lack the ability to edit a generated image through textual prompt(s) at all or lack the ability to edit a generated image through textual prompt(s) exclusively.

[0006] Moreover, some proposed LLI-based editing methods may not enable editing of a real image (*e.g.*, captured by a real-world physical camera). For example, some methods may not enable starting with a real image, that includes a basket with apples, and a natural language (NL) caption that is descriptive of the image (*e.g.*, a human provided prompt of "a basket with apples") –and generating a synthetic version of the real image to include a basket with cookies (in lieu of apples) in response to an edit, to the NL caption, to replace "apples" with "cookies". For instance, to enable such editing utilizing LLI-based techniques, the real image must be inverted, in view of the corresponding LLI and the NL caption, to enable generating a synthetic version of the real image using the LLI. Put another way, to enable such editing, the real image must be inverted in a manner such that the LLI model can then be utilized to process the inversion, and an edited NL caption, to generate a synthetic image that is visually similar to the real image, but that includes visual modifications consistent with one or more edits that are reflected by the edited NL caption.

Summary

[0007] Implementations of the present disclosure are directed to enabling editing of a real image via editing only an NL caption of the real image (also referred to herein as prompt-to-prompt editing). This enables voice-based, typed (*e.g.*, physical or virtual keyboard), and/or touch-based (*e.g.*, interaction with an emphasis element, selection of alternative term(s)) input to edit a real image, and obviates the need for any specification of an image mask and/or other input(s). Such inputs for editing are natural, can be made with low latency, and enable various editing tasks that are challenging otherwise. Further, implementations disclosed herein do not require extra, and computationally expensive, model training, fine-tuning, extra data, or optimization.

[0008] More particularly, some implementations are directed to generating a synthetic image, that corresponds to a real image captured using a real camera, but that is generated based on,

and reflects, edit(s) to an original natural language (NL) caption for the real image. In some of those implementations, the original NL caption is used in performing an inversion on the real image to generate a diffusion trajectory for the real image. Further, the diffusion trajectory is used to optimize a sequence of unconditional embeddings, for the real image, that are not based on the NL caption for the real image. Yet further, the edited NL caption, the unconditional embeddings, and at least part of the noise vector can be processed, using a Large-scale language-image (LLI) model, to generate the synthetic image.

[0009] As a non-limiting example, assume a user provided or automatically generated NL caption for a real image is "a furry bear watching a bird", and the real image reflects a furry bear that is watching a bird. The edit(s) to the NL caption can include a replacement of a subset of tokens of the source NL prompt with replacement token(s) (*e.g.*, replacing "bird" with "butterfly"), an addition of token(s) to the source NL prompt (*e.g.*, adding "blue" before "bird"), and/or an adjustment of emphasis on token(s) of the source NL prompt (*e.g.*, increasing emphasis on "fuzzy"). Implementations disclosed herein can utilize such edit(s) to generate a synthetic image, that corresponds to the real image, but that reflects the edit(s). For example, replacing "bird" with "butterfly" can result in generation of a synthetic image that is visually similar to the real image, but that includes the bear watching a butterfly (in lieu of a bird).

[0010] It should be appreciated that all combinations of the foregoing concepts and additional concepts described in greater detail herein are contemplated as being part of the subject matter disclosed herein. For example, all combinations of claimed subject matter appearing at the end of this disclosure are contemplated as being part of the subject matter disclosed herein.

Brief Description of the Drawings

[0011] FIG. 1 schematically depicts example components and interactions that can be utilized in implementations disclosed herein.

[0012] FIG. 2 illustrates an example method of inverting a real image in view of an NL caption for the real image, including generating and storing, for the real image, a final noise vector and an optimized sequence of unconditional embeddings.

[0013] FIG. 3 illustrates an example method of generating, using an LLI model, an edited image that is visually similar to a real image, but that includes visual modifications consistent with an edit to an NL caption for the real image.

[0014] FIG. 4 schematically illustrates an example computer architecture on which selected aspects of the present disclosure can be implemented.

Detailed Description

[0015] Prior to turning to the figures, a non-limiting overview of various implementations is provided.

[0016] The progress in image synthesis using LLIs, such as text-guided diffusion models, has attracted much attention due to their exceptional realism and diversity. LLIs have ignited the imagination of multitudes of users, enabling image generation with unprecedented creative freedom. Naturally, this has initiated ongoing research efforts, investigating how to harness these powerful models for image editing. For example, intuitive text-based prompt-to-prompt editing has been recently demonstrated over synthesized images, allowing a user to easily manipulate a synthesized image using text only.

[0017] However, text-guided editing of a real image requires inverting the given image and a textual prompt. That is, finding an initial noise vector that produces the input image when fed with the textual prompt into the diffusion process, while preserving the editing capabilities of the model. A deterministic denoising diffusion implicit model (DDIM) inversion scheme has been suggested for unconditional diffusion models (*i.e.*, diffusion models not condition on a textual prompt). However, DDIM can have shortcomings for text-guided diffusion models when classifier-free guidance, which is necessary for meaningful prompt-to-prompt editing, is applied.

[0018] In view of these and other considerations, implementations disclosed herein relate to an effective inversion scheme, achieving near-perfect reconstruction, while retaining the rich text-guided editing capabilities of the original model. Those implementations focus on two aspects of guided diffusion models: classifier-free guidance and DDIM inversion.

[0019] In classifier-free guidance, in each diffusion step, the prediction is performed twice:

once unconditionally and once with the text condition. These predictions are then extrapolated to amplify the effect of the text guidance. While the conditional prediction is impactful, implementations disclosed herein also recognize the substantial effect induced by the unconditional part. Accordingly, some of those implementations optimize the embedding used in the unconditional part in order to invert the input image and a text prompt. This is also referred to herein as null-text optimization, as the embedding of the empty text string is replaced with an optimized embedding.

[0020] DDIM inversion includes performing DDIM sampling in reverse order. Although a slight error is introduced in each step, this works well in the unconditional case. However, in practice, it breaks for text-guided image synthesis since classifier-free guidance magnifies its accumulated error. Nonetheless, implementations disclosed herein recognize that DDIM inversion offers a promising starting point for the inversion. Those implementations use the sequence of noised latent codes, obtained from an initial DDIM inversion, as a pivot and then perform an optimization around this pivot to yield an improved and more accurate inversion. This is also referred to herein as diffusion pivotal inversion, which stands in contrast to some works that aim to map all possible noise vectors to a single image.

[0021] Large-scale diffusion models have recently raised the bar for the task of generating images conditioned on plain text, known as text-to-image synthesis. Exploiting the powerful architecture of diffusion models, these models can generate practically any image by simply feeding a corresponding text to a diffusion model, and have changed the landscape of artistic applications.

[0022] However, synthesizing very specific or personal objects which are not widespread in the training data has been challenging. This requires an inversion process that, given an input image, would enable regenerating the depicted object(s) using a text-guided diffusion model. Inversion has been studied for GANs, ranging from latent-based optimization and encoders to feature space encoders and fine-tuning of the model. However, prior works struggle to edit a given real image while accurately reproducing the unedited parts.

[0023] Other prior works have attempted to adapt text-guided diffusion models to the fundamental challenge of single-image editing, aiming to exploit their rich and diverse semantic knowledge. However, those prior works struggle to accurately preserve the input

image details. To overcome this, those prior works assume that the user provides a mask to restrict the region in which the changes are applied, achieving both meaningful editing and background preservation. However, requiring that users provide a precise mask is burdensome, requiring multiple user inputs. Further, masking the image content removes important information, which is mostly ignored in the inpainting process.

[0024] As an overview of some implementations disclosed herein, let I be a real image. Implementations disclosed herein seek to edit I, using only text guidance, to get an edited image $I*$, where the editing is guided by a source NL caption $\mathcal{P}$ and edited NL caption $\mathcal{P}*$. The source NL caption $\mathcal{P}$ can be, for example, user provided or automatically generated using, for instance, a visual language model (VLM) or an off-the-shelf captioning model. For example, given a real image I of a baby wearing a blue shirt and lying on a sofa and a source NL caption $\mathcal{P}$ of "A baby wearing a blue shirt lying on the sofa", an edited image $I*$ that replaces the baby can be generated based on an edited NL caption $\mathcal{P}*$ that replaces "baby" with "robot" (*i.e.*, "A robot wearing a blue shirt lying on the sofa").

[0025] Such editing operations first require inverting I to the model's output domain. Namely, inverting I so that I can be reconstructed by feeding the inversion, from the inverting, and the source prompt $\mathcal{P}$ to the model, while still retaining the intuitive text-based editing abilities.

[0026] Implementations disclosed herein recognize that DDIM inversion produces unsatisfying reconstruction when classifier-free guidance is applied, but provides a good starting point for an optimization, enabling efficient achievement of a high-fidelity inversion. Implementations also recognize that optimizing the unconditional null embedding, which is used in classifier-free guidance, allows an accurate reconstruction while avoiding the tuning of the model and the conditional embedding – thereby preserving the desired editing capabilities.

[0027] Text-guided diffusion models aim to map a random noise vector z_t and textual condition $\mathcal{P}$ to an output image z_0, where $\mathcal{P}$ corresponds to the given conditioning prompt. In order to perform sequential denoising, the network ϵ_0 is trained to predict artificial noise, following the objective:

$$\min_{\theta} E_{z_0,\varepsilon\sim N(0,I),t\sim \mathrm{Uniform}(1,T)}\|\epsilon-\epsilon_\theta(z_t,t,C)\|_2^2. \quad (1)$$

[0028] Note that $C = \psi(\mathcal{P})$ is the embedding of the text condition and z_t is a noised sample,

where noise is added to the sampled data z_0 according to timestamp t. At inference, given a noise vector z_T, the noise is gradually removed by sequentially predicting it using the trained network for T steps.

[0029] Since the aim is to accurately reconstruct a given real image, deterministic DDIM sampling can be employed:

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t} z_t + \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) . \varepsilon_\theta(z_t, t, C)}.$$

[0030] Diffusion models often operate in the image pixel space where z_0 is a sample of a real image. For example, some diffusion models operate such that the diffusion forward process is applied on a latent image encoding $z_0 = E(x_0)$, and an image decoder is employed at the end of the diffusion backward process $x_0 = D(z_0)$.

[0031] One of the challenges in text-guided generation is the amplification of the effect induced by the conditioned text. To this end, classifier-free guidance techniques have been proposed, where the prediction is also performed unconditionally, which is then extrapolated with the conditioned prediction. More formally, let $\emptyset = \psi("")$ be the embedding of a null text and let ω be the guidance scale parameter (*e.g.*, 7.5 or other parameter, such as a default parameter), then the classifier-free guidance prediction is defined by:

$\varepsilon_\theta(z_t, \widetilde{t, C, \emptyset}) = \omega . \varepsilon_\theta(z_t, t, C) + (1 - \omega) . \varepsilon_\theta(z_t, t, \emptyset)$.

[0032] With DDIM inversion, there is an assumption that the ODE process can be reversed in the limit of small steps:

$$z_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t} z_t + \left(\sqrt{\frac{1}{\alpha_{t+1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) . \varepsilon_\theta(z_t, t, C)}.$$

[0033] In other words, with DDIM inversion the diffusion process is performed in the reverse direction, that is $z_0 \rightarrow z_T$ instead of $z_T \rightarrow z_0$, where z_0 is set to be the encoding of the given real image.

[0034] Recent inversion works use random noise vectors for each iteration of their

optimization, aiming at mapping every noise vector to a single image. Implementations disclosed herein recognize that this is inefficient as inference requires only a single noise vector. Instead, those implementations seek to perform a more local optimization, such as one that uses only a single noise vector. For example, some of those implementations aim to perform an optimization around a pivotal noise vector which is a good approximation and thus allows a more efficient inversion.

[0035] With DDIM inversion, a slight error is incorporated in every step. For unconditional diffusion models, the accumulated error is negligible and the DDIM inversion succeeds. However, meaningful editing using some text-guided diffusion models requires applying classifier-free guidance with a large guidance scale (*e.g.*, $w > 1$). Such a guidance scale amplifies the accumulated error. Therefore, performing the DDIM inversion procedure with classifier-free guidance results not only in visual artifacts, but the obtained noise vector might be out of the Gaussian distribution. The latter decreases the editability. That is, the latter decreases the ability to edit using the particular noise vector.

[0036] Using DDIM inversion with guidance scale $w = 1$ provides a rough approximation of the original image, which is highly editable but far from accurate. More specifically, the reversed DDIM produces a *T* steps trajectory between the image encoding z_0 to a Gaussian noise vector z_T^*. Again, a large guidance scale is essential for editing. Hence, implementations focus on providing z_T^* to the diffusion process with classifier-free guidance (*e.g.*, $w > 1$). This results in high editability but inaccurate reconstruction, since the intermediate latent codes deviate from the trajectory.

[0037] Motivated by the high editability, this initial DDIM inversion with $w = 1$ is referred to herein as a pivot trajectory and optimization is performed around the pivot trajectory with the guidance scale, $w > 1$. That is, the optimization maximizes the similarity to the original image, while maintaining the ability to perform meaningful editing. A separate optimization can be performed for each timestamp t in the order of the diffusion process $t = T \rightarrow t = 1$, with the objective of getting as close as possible to the initial trajectory:

$$z_T^*, \ldots, z_0^*: \min \| z_{t-1}^* - z_{t-1} \|_2^2, \qquad (2)$$

[0038] In the preceding, z_{t-1} is the intermediate result of the optimization. Since pivotal

DDIM inversion provides a good starting point, this optimization is highly efficient compared to using random noise vectors. Note that for every $t < T$, the optimization should start from the endpoint of the previous step $(t + 1)$ optimization, otherwise the optimized trajectory would not hold at inference. Accordingly, after the optimization of step t, the current noisy latent code $\bar{z}_t$ can be computed, which is then used in the optimization of the next step to ensure the new trajectory would end near z_0 (see *e.g.*, Eq. (3) for more details).

[0039] To successfully invert real images into a domain of a text-guided diffusion model, some works optimize the textual encoding, the model's weights, or both. Fine-tuning the model's weight for each image involves duplicating the entire model, which is highly inefficient in terms of memory consumption. Moreover, unless fine-tuning is applied for each and every edit, it necessarily hurts the learned prior of the model and therefore the semantics of the edits. Direct optimization of the textual embedding results in a non-interpretable representation since the optimized tokens do not necessarily match pre-existing words. Therefore, an intuitive prompt-to-prompt edit becomes more challenging with such works.

[0040] Instead, implementations disclosed herein exploit a feature of classifier-free guidance. That is, that the result is highly affected by the unconditional prediction. Those implementations replace the default null-text embedding with an optimized one, referred to herein as null-text optimization. Namely, for each input image, only the unconditional embedding $\emptyset$ is optimized, where the unconditional embedding $\emptyset$ can be initialized with the null-text embedding. The model and the conditional textual embedding are kept unchanged.

[0041] This results in high-quality reconstruction while still allowing intuitive editing (*e.g.*, prompt-to-prompt editing) by simply using the optimized unconditional embedding. Moreover, after a single inversion process, the same unconditional embedding can be used for multiple editing operations over the input image. This is computationally efficient as only a single inversion is needed for multiple editing operations. Since null-text optimization is naturally less expressive than fine-tuning the entire model, it requires the more efficient pivotal inversion scheme.

[0042] Implementations disclosed herein refer to optimizing a single unconditional embedding $\emptyset$ as a global null-text optimization. However, some implementations recognize that optimizing a different "null embedding" $\emptyset_t$ for each timestamp t significantly improves the

reconstruction quality, which is well suited for pivotal inversion disclosed herein. Accordingly, those implementations use per-timestamp unconditional embeddings $\{\emptyset_t\}_{t=1}^{T}$, and initialize $\emptyset_t$ with the embedding of the previous step $\emptyset_{t+1}$. Algorithm 1 is presented below and provides an example of some of those implementations.

Algorithm 1: Null-text inversion

---

1 Input: A source prompt embedding $C = \psi(\mathcal{P})$ and Input image I;
2 Output: Noise vector z_T and optimized embeddings $\{\emptyset_t\}_{t=1}^{T}$.

---

3 Set guidance scale $w = 1$;
4 Compute the intermediate results $z_T^*, \ldots, z_0^*$ using DDIM inversion over *I;*
5 Set guidance scale $w = 7.5$;
6 Initialize $z_T^- \leftarrow z_T^*, \emptyset_T \leftarrow \psi(\text{""})$;
7 for $t = T, T-1, \ldots, 1$ **do**
8 | **for** $j = 0, \ldots, N-1$ **do**
9 | | $\phi_t \leftarrow \phi_t - \eta \nabla_\emptyset \| z_{t-1}^* - z_{t-1}(\overline{z}_t, \emptyset_t, C) \|_2^2$;
10 | **end**
11 | Set $\overline{z_{t-1}} \leftarrow z_{t-1}(\overline{z}_t, \emptyset_t, C), \emptyset_{t-1} \leftarrow \emptyset_t$;
12 end
13 Return $\overline{z_T}, \{\emptyset_t\}_{t=1}^{T}$

[0043] The DDIM inversion, with $w = 1$, outputs a sequence of noisy latent codes $z_T^* \ldots, z_0^*$ where $z_0^* = z_0$. $z_T^- = z_t$ can be initialized and the following optimization performed with a guidance scale (*e.g.*, 7.5 or other default guidance scale) for the timestamps $t = T, \ldots, 1$, each for *N* iterations:

$$\min_{\emptyset_t} \| z_{t-1}^* - z_{t-1}(z_t^-, \emptyset_t, C) \|_2^2. \qquad (3)$$

[0044] For simplicity, $z_{t-1}(\overline{z}_t, \emptyset_t, C)$ denotes applying DDIM sampling step using $\overline{z}_t$, the unconditional embedding $\emptyset_t$, and the conditional embedding C. At the end of each step,

$\overline{z_{t-1}} = z_{t-1}(\bar{z}_t, \emptyset_t, C)$ is updated. Some implementations implement early stopping, which reduces time and/or processor resource consumption. Finally, the real input image can be edited by using the noise $\overline{z_T}$=z_T^* and the optimized unconditional embeddings $\{\emptyset_t\}_{t=1}^{T}$.

[0045] Accordingly, implementations provide an approach to invert real images, with corresponding captions, into the latent space of a text-guided diffusion model while maintaining its powerful editing capabilities. Some of those implementations first use DDIM inversion to compute a sequence of noisy latent codes, which roughly approximate the original image (with the given caption), then use this sequence as a fixed pivot to optimize the input null-text embedding. Such optimization compensates for the inevitable reconstruction error caused by the classifier-free guidance component. Once the image-caption pair is accurately embedded in the output domain of the model, prompt-to-prompt editing can be instantly applied at inference time. Through utilization of pivotal inversion and null-text optimization, implementations bridge the gap between reconstruction and editability of real images. Further, implementations avoid computationally intensive model-tuning.

[0046] The algorithm for optimizing only a single null-text embedding Ø for all timestamps is presented below in algorithm 2. In this case, since the optimization of Ø in a single timestamp affects all other timestamps, the order of the iterations is changed relative to Algorithm 1. That is, *N* iterations are performed and, in each, Ø is optimized for all the diffusion timestamps by iterating over *t*.

Algorithm 2: Global-null-text inversion

---

1 Input: A source prompt P and input image *I*.
2 Output: Noise vector z_T and an optimized embeddings Ø.

---

3 Set guidance scale $w = 1$;
4 Compute the intermediate results $z_T^*, \ldots, z_0^*$ of DDIM inversion for image *I;*
5 Set guidance scale $w = 7.5$;
6 Initialize $\emptyset \leftarrow \psi$ ("");
7 for $j = 0, \ldots, N-1$ **do**

8 | Set $\overline{z_T} \leftarrow z_T^*$;
9 | **for** $t = T, T-1, \ldots, 1$ **do**
10 | $\emptyset \leftarrow \emptyset - \eta \nabla \emptyset \parallel z_{t-1}^* - z_{t-1}(\overline{z_t}, \emptyset, C) \parallel_2^2$;
Set $\overline{z_{t-1}} \leftarrow z_{t-1}(\overline{z_t}, \emptyset, C)$;
11 | **end**
12 end
13 Return $\overline{z_T}, \emptyset$

[0047] Diffusion Denoising Probabilistic Models (DDPM) are generative latent variable models that aim to model a distribution $p_\theta(x_\theta)$ that approximates the data distribution $q(x_\theta)$ and that are easy to sample from. DDPMs model a "forward process" in the space of x_θ from data to noise. This is called "forward" due to its procedure progressing from x_θ to x_T. Note that this process is a Markov chain starting from x_θ, where noise is gradually added to the data to generate the latent variables $x_{1,\ldots,}x_T \in X$. The sequence of latent variables, therefore, follows $q(x_1, \ldots, x_t|x_0) = \prod_{i=1}^{t} q(x_t|x_{t-1})$, where a step in the forward process is defined as a Gaussian transition $q(x_t|x_{t-1}) \coloneqq N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ parameterized by a schedule $\beta_0, \ldots, \beta_T \in (0,1)$. When *T* is large enough, the last noise vector x_T nearly follows an isotropic Gaussian distribution.

[0048] An interesting property of the forward process is that one can express the latent variable x_0 directly as the following linear combination of noise and x_0 without sampling intermediate latent vectors:

$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}w_{\sim}N(0, I)$, (5), where $a_t \coloneqq \prod_{i=1}^{t}(1-\beta_i)$.

[0049] To sample from the distribution $q(x_0)$, the dual "reverse process" $p(x_{t-1}|x_t)$ can be defined from isotropic Gaussian noise x_T to data by sampling the posteriors $q(x_{t-1}|x_t)$. Since the intractable reverse process $q(x_{t-1}|x_t)$ depends on the unknown data distribution $q(x_0)$, it can be approximated with a parameterized Gaussian transition network $p_\theta(x_{t-1}|x_t) \coloneqq N(x_{t-1}|\mu_\theta(x_t, \sum_\theta(x_t, t))$. The $\mu_\theta(x_t, t)$ can be replaced by predicting the noise $\epsilon_\theta(x_t, t)$ added to x_0 using equation 5.

[0050] Bayes' theorem can be used to approximate:

$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\varepsilon_\theta(x_t, t))$. (6)

[0051] Once there is a trained $\epsilon_\theta(x_t, t)$, the following sample method can be used:

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z, z \sim N(0, I). \quad (7)$$

[0052] The σ_t of each sample stage can be controlled, and in DDIMs the sampling process can be made deterministic using σ_t in all the steps. The reverse process can finally be trained by solving the following optimization problem:

$\min_\theta L(\theta) := \min_\theta E_{x_0 \sim q(x_0), w \sim N(0,I), t} \|w - \varepsilon_\theta(x_t, t)\|_2^2$, teaching the parameters θ to fit $q(x_0)$ by maximizing a variational lower bound.

[0053] As a working example of various implementations of prompt-to-prompt editing disclosed herein, let *I* be a source image that is a real image and let *P* be an NL caption for the real image. Some implementations seek to edit the source image *I* guided only by an edited NL caption *P**, resulting in an edited image *I**. For example, consider a source image *I* that is a picture of a new bicycle and that includes the NL caption *P* "my new bicycle", and assume that the user wants to edit the color of the bicycle, its material, or even replace it with a scooter while preserving the appearance and structure of the source image *I*. An intuitive interface for the user is to directly change the NL caption *P* by further describing the appearance of the bike (*e.g.*, adding "green" before "bicycle"), or replacing it with another word (*e.g.*, replacing "bicycle" with "scooter"). Various implementations disclosed herein avoid relying on any user-defined mask (*e.g.*, defined through interaction with the source image *I*) to assist or signify where the edit to the source image *I* should occur. For example, those various implementations avoid relying on any user-defined mask that defines the "bicycle" in the source image *I* and that is generated based on user interaction with the source image *I*.

[0054] Rather, implementations disclosed herein can use the inversion of the real image, and the edited NL caption, in generating an edited synthetic image that is visually similar to the real image, but that includes the edit(s) reflected in the edited NL caption. For example, in generating the edited synthetic image implementations can process, using an LLI model, the edited NL caption and the final noise vector, and the optimized sequence of unconditional embeddings from inversion of the real image. During the processing, edited conditional embedding(s), that are conditioned based on the edited NL caption, can be generated.

[0055] Some examples of specific editing operations, that can be used to define an edited NL

caption, are now provided. Those examples include word swap (also referred to as replacement), adding a new phrase (also referred to as addition), and attention reweighting (also referred to as emphasis adjustment).

[0056] With word swap, user interface input is provided that indicates a user has swapped token(s) of the original NL caption with others. For example, "bicycle" can be swapped for "car" when the user interface input indicates an edit of the original NL caption of "a big red bicycle" to an edited NL caption of "a big red car". Such user interface input can be via touch and/or typed inputs to delete "bicycle" and type "car" and/or via spoken user interface input (*e.g.*, spoken input of "replace bicycle with car").

[0057] With adding a new phrase, user interface input is provided that indicates a user has added new token(s) to the original NL caption. For example, "children drawing of" can be prepended to an original NL caption of "a castle next to a river", when the user interface input indicates such prepending. For example, the user interface input can include typing "children drawing of" at the beginning of the original NL caption or can be spoken user interface input such as "prepend children drawing of".

[0058] With attention re-weighting, user interface input is provided that indicates a user desire to strengthen or weaken the extent to which token(s) of the original NL caption are affecting the real image. For example, the original NL caption can be "a fluffy red ball", and the user may want an edited image where the ball is more fluffy or less fluffy than it is in the real image. User interface input that indicates such an increase or decrease in fluffiness can be, for example, interaction with a slider or up and down arrows that are presented in conjunction with "fluffy", bolding or underlining "fluffy", and/or spoken input (*e.g.*, "more fluffy").

[0059] Turning now to the Figures, FIG. 1 schematically depicts example components and interactions that can be utilized in implementations disclosed herein. For example, components and interactions that can be involved in generating, using the LLI model 150, an edited image 106 that is visually similar to a real image 102, but that includes visual modifications consistent with an edit to an NL caption 101 for the real image 102, where the edit is reflected in caption edit input 105.

[0060] In FIG. 1, a client device 110 can provide a real image 102 to a diffusion trajectory

engine 120. The diffusion trajectory engine 120 can generate a diffusion trajectory 103 for the real image 102. For example, the diffusion trajectory engine 120 can generate the diffusion trajectory 103 using an inversion process and the real image, such as a DDIM inversion process. The diffusion trajectory 103 can include an encoding of the real image 102, a final noise vector, and a sequence of intermediate noise vectors between the encoding and the final noise vector.

[0061] The diffusion trajectory 103 is provided to the optimization engine 125, along with an NL caption 101 for the real image 102. The NL caption 101 can be provided by the client device 110 and based on user interface input (*e.g.*, user interface input that is a user-curated caption for the real image 102) and/or can be provided by a caption engine 140 that automatically generates the NL caption 101 by processing the real image 102 using a caption model or other machine learning model(s).

[0062] The optimization engine 125 generates an optimized sequence of unconditional embeddings 104. In generating the optimized sequence of unconditional embeddings 104, the optimization engine 125 can generate the optimized sequence of unconditional embeddings 104 based on comparisons of generated predicted latent codes to the diffusion trajectory 103, where each of the generated predicted latent codes is generated during a respective time step of processing, using LLI model 150, that is based on a conditional embedding that is conditioned based on the NL caption 101.

[0063] The edited image engine 130 receives caption edit input 105, that is user interface input provided at the client device 110 and that specifies one or more edits to the NL prompt 101 (which can be rendered at the client device 110 – optionally based on output from the caption engine 140), such as replacement input, addition input, and/or emphasis adjustment input. In response to receiving the caption edit input 105, the edited image engine 130 can interact with the LLI model 150 in generating an edited image 107 that is visually similar to the real image 102, but that includes visual modifications that are consistent with the edit(s), to the NL caption 101, that are reflected by the caption edit input 105.

[0064] In interacting with the LLI model 150 in generating the edited image 106, the edited image engine 130 can utilize the caption edit input 105, at least part of the diffusion trajectory, and the optimized sequence of unconditional embeddings 104.

[0065] FIG. 2 illustrates an example method 200 of inverting a real image in view of an NL

caption for the real image, including generating and storing, for the real image, a final noise vector and an optimized sequence of unconditional embeddings. For convenience, the operations of the flow chart are described with reference to a system that performs the operations. This system can include various components of various computer systems, such as one or more components of server computing device(s). Moreover, while operations of method 200 are shown in a particular order, this is not meant to be limiting. One or more operations can be reordered, omitted or added.

[0066] At block 202, the system identifies a real image captured by a real camera, such a real image uploaded from a client device.

[0067] At block 204, the system identifies an NL caption for the real image. In identifying the NL caption for the real image, the system can perform sub-block 204A or sub-block 204B.

[0068] At sub-block 204A, the NL caption for the real image is generated based on user interface input. For example, when the real image is received from the client device at block 202, the NL prompt can also be received and can be responsive to user interface input received at the client device. For instance, the user interface input can be received, at the client device, responsive to rendering a prompt such as a prompt of "please provide a natural language description of this image".

[0069] At sub-block 204B, the NL prompt for the real image is generated based on processing the real image using a captioning model or other visual language model.

[0070] At block 206, the system performs an inversion on the real image to generate a diffusion trajectory. The diffusion trajectory can include an encoding of the real image, a final noise vector, and a sequence of intermediate noise vectors (e.g., between the final noise vector and the encoding). In some implementations, in performing the inversion, the system uses a first guidance scale for classifier-free guidance of the text-guided diffusion model, such as a first guidance scale that causes unconditional embeddings to have a lesser (*e.g.*, no or de minimis) impact in text-guided diffusion. In some implementations, in performing the inversion, the system uses a DDIM inversion process, such as one that includes performing DDIM sampling in reverse order and starting with the encoding of the real image.

[0071] At block 208, the system optimizes a sequence of unconditional embeddings based on comparisons of generated predicted latent codes to the diffusion trajectory of block 206. In

some implementations, the system generates each of the predicted latent codes during a respective time step of processing, using a text-guided diffusion model, that is based on a conditional embedding that is conditioned based on the NL caption for the real image. In some implementations, in optimizing the sequence of unconditional embeddings the system uses, during processing using the text-guided diffusion model, a second guidance scale for classifier-free guidance of the text-guided diffusion model, such as a second guidance scale that impacts text-guided diffusion (*e.g.*, at least to a greater extent than does a guidance scale utilized in block 206). In some implementations, the comparisons of the generated predicted latent codes to the diffusion trajectory for the image each include a respective mean squared error.

[0072] At block 210, the system stores (*e.g.*, at least temporarily in memory), at least the final noise vector and the optimized sequence of unconditional embeddings. The system can store the final noise vector and the optimized sequence of unconditional embeddings in association with the real image of block 202.

[0073] FIG. 3 illustrates an example method 300 of generating, using an LLI model, an edited image that is visually similar to a real image, but that includes visual modifications consistent with an edit to an NL caption for the real image. For convenience, the operations of the flow chart are described with reference to a system that performs the operations. This system can include various components of various computer systems, such as one or more components of server computing device(s). Moreover, while operations of method 300 are shown in a particular order, this is not meant to be limiting. One or more operations can be reordered, omitted or added.

[0074] At block 302, the system obtains an edited NL caption that is based on user interface input and that is an edited version of an NL caption for the real image (*e.g.*, an edited version NL caption of block 204 of method 200). The edited NL caption can be generated based on user interface input received, at a client device, responsive to rendering of the real image and/or responsive to rendering the NL caption for the source image. The real image can be the real image of block 202 of an iteration of method 200 of FIG. 2.

[0075] In various implementations, block 302 includes one or more of sub-blocks 302A, 302B, and 302C. At sub-block 302A, the user interface of block 302 input includes replacement input. The replacement input can reflect an edit that is a replacement, of a subset of tokens of the NL

caption, with one or more replacement tokens that differ from the subset of tokens of the NL caption. At sub-block 302B, the user interface of block 302 input includes addition input. The addition input can reflect an edit that is an addition, of one or more additional tokens, to the NL caption. At sub-block 302A, the user interface of block 302 input includes emphasis adjustment input. The emphasis adjustment input can reflect an edit that is an adjustment of emphasis on one or more emphasis tokens of the NL caption, where the adjustment is an increase or decrease of emphasis and can optionally reflect a magnitude of the increase or decrease.

[0076] At block 304 the system obtains, for the real image, at least a final noise vector and an optimized sequence of unconditional embeddings. For example, the system can obtain, for the real image, a final noise vector and an optimized sequence of unconditional embeddings stored, at block 210 of method 200, based on processing the real image and the NL caption (unedited) using method 200.

[0077] At block 306, the system generates an edited image based on processing, using an LLI model, the final noise vector, the optimized sequence of unconditional embeddings, and an edited conditional embedding that is conditioned based on the edited NL caption. In some implementations, in generating the edited image, the system uses the edited conditional embedding in each of a plurality of time steps of the processing, and uses a corresponding next one of the unconditional embeddings in each of the plurality of time steps of the processing. In some implementations, in generating the edited image using the LLI model, the system uses the LLI model without tuning any weights of the text-guided diffusion model.

[0078] At block 308, the system causes rendering of the edited image and, optionally, of the edited NL caption. For example, the system can cause such rendering at a client device that provided the user interface input of block 302.

[0079] At optional block 310, the system can monitor for new user interface input that indicates a further edit to the NL caption, and that is in addition to edit(s) of prior iteration(s) of block 302. If such new user interface input is detected, the system can proceed to perform another iteration of block 302, 304, 306, and 308 based on such new user interface input.

[0080] FIG. 4 is a block diagram of an example computing device 410 that can optionally be utilized to perform one or more aspects of techniques described herein. For example, all or

aspects of computing device 410 can be incorporated in server(s) or other computing device(s) that are utilized to implement prompt-to-prompt editing techniques disclosed herein.

[0081] Computing device 410 typically includes at least one processor 414 which communicates with a number of peripheral devices via bus subsystem 412. These peripheral devices can include a storage subsystem 424, including, for example, a memory subsystem 424 and a file storage subsystem 426, user interface output devices 420, user interface input devices 422, and a network interface subsystem 416. The input and output devices allow user interaction with computing device 410. Network interface subsystem 416 provides an interface to outside networks and is coupled to corresponding interface devices in other computing devices.

[0082] User interface input devices 422 can include a keyboard, pointing devices such as a mouse, trackball, touchpad, or graphics tablet, a scanner, a touch screen incorporated into the display, audio input devices such as voice recognition systems, microphones, and/or other types of input devices. In general, use of the term "input device" is intended to include all possible types of devices and ways to input information into computing device 410 or onto a communication network.

[0083] User interface output devices 420 can include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices. The display subsystem can include a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), a projection device, or some other mechanism for creating a visible image. The display subsystem can also provide non-visual display such as via audio output devices. In general, use of the term "output device" is intended to include all possible types of devices and ways to output information from computing device 410 to the user or to another machine or computing device.

[0084] Storage subsystem 424 stores programming and data constructs that provide the functionality of some or all of the modules described herein. For example, the storage subsystem 424 can include the logic to perform selected aspects of the methods of FIGS. 2 and/or 3, as well as to implement various components described herein.

[0085] These software modules are generally executed by processor 414 alone or in combination with other processors. Memory 424 used in the storage subsystem 424 can

include a number of memories including a main random-access memory (RAM) 430 for storage of instructions and data during program execution and a read only memory (ROM) 432 in which fixed instructions are stored. A file storage subsystem 426 can provide persistent storage for program and data files, and can include a hard disk drive, a floppy disk drive along with associated removable media, a CD-ROM drive, an optical drive, or removable media cartridges. The modules implementing the functionality of certain implementations can be stored by file storage subsystem 426 in the storage subsystem 424, or in other machines accessible by the processor(s) 414.

[0086] Bus subsystem 412 provides a mechanism for letting the various components and subsystems of computing device 410 communicate with each other as intended. Although bus subsystem 412 is shown schematically as a single bus, alternative implementations of the bus subsystem can use multiple busses.

[0087] Computing device 410 can be of varying types including a workstation, server, computing cluster, blade server, server farm, or any other data processing system or computing device. Due to the ever-changing nature of computers and networks, the description of computing device 410 depicted in Fig. 4 is intended only as a specific example for purposes of illustrating some implementations. Many other configurations of computing device 410 are possible having more or fewer components than the computing device depicted in FIG. 4.

[0088] While several implementations have been described and illustrated herein, a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein can be utilized, and each of such variations and/or modifications is deemed to be within the scope of the implementations described herein. More generally, all parameters, dimensions, materials, and configurations described herein are meant to be exemplary and that the actual parameters, dimensions, materials, and/or configurations will depend upon the specific application or applications for which the teachings is/are used. Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific implementations described herein. It is, therefore, to be understood that the foregoing implementations are presented by way of example only and that, within the scope of the appended claims and

equivalents thereto, implementations can be practiced otherwise than as specifically described and claimed. Implementations of the present disclosure are directed to each individual feature, system, article, material, kit, and/or method described herein. In addition, any combination of two or more such features, systems, articles, materials, kits, and/or methods, if such features, systems, articles, materials, kits, and/or methods are not mutually inconsistent, is included within the scope of the present disclosure.

[0089] In some implementations, a method implemented by processor(s) is provided and includes identifying a real image captured by a real camera and identifying a natural language (NL) caption for the real image. The method further includes performing an inversion on the real image to generate a diffusion trajectory for the real image. The diffusion trajectory includes an encoding of the real image, a final noise vector, and a sequence of intermediate noise vectors between the encoding and the final noise vector. The method further includes, subsequent to generating the diffusion trajectory for the real image, optimizing a sequence of unconditional embeddings that are not based on the NL caption for the real image. Optimizing the sequence of unconditional embeddings can be based on comparisons of generated predicted latent codes to the diffusion trajectory for the real image. Each of the generated predicted latent codes can be generated during a respective time step of processing, using a text-guided diffusion model, that is based on a conditional embedding that is conditioned based on the NL caption for the real image. The method further includes obtaining an edited NL caption that is an edited version of the NL caption for the real image. The method further includes generating an edited image that is visually similar to the real image but that includes visual modifications consistent with one or more edits that are reflected by the edited NL caption. Generating the edited image can include generating the edited image based on processing, using the text-guided diffusion model: the final noise vector, the optimized sequence of unconditional embeddings, and an edited conditional embedding that is conditioned based on the edited NL caption.

[0090] These and other implementations disclosed herein can include one or more of the following features. In some implementations, performing the inversion includes using, during the inversion a first guidance scale, for classifier-free guidance of the text-guided diffusion model – and optimizing the sequence of unconditional embeddings includes using, during

processing using the text-guided diffusion model, a second guidance scale for classifier-free guidance of the text-guided diffusion model. In some versions of those implementations, the first guidance scale causes the unconditional embeddings to have a lesser impact, in text-guided diffusion, than does the second guidance scale. For example, the first guidance scale can cause the unconditional embeddings to have no, or de minimis, impact in text-guided diffusion. In some of those versions, generating the edited image includes using the second guidance scale during processing, using the text-guided diffusion model, to generate the edited image.

[0091] In some implementations, generating the edited image based on processing, using the text-guided diffusion model: the final noise vector, the optimized sequence of unconditional embeddings, and the edited conditional embedding includes: using the edited conditional embedding in each of a plurality of time steps of the processing; and using a corresponding next one of the unconditional embeddings in each of the plurality of time steps of the processing.

[0092] In some implementations, performing the inversion on the real image includes performing the inversion using a deterministic denoising diffusion implicit model (DDIM) inversion process. In some of those implementations, performing the inversion using the DDIM inversion process includes performing DDIM sampling in reverse order and starting with the encoding of the real image.

[0093] In some implementations, the comparisons of the generated predicted latent codes to the diffusion trajectory for the image each include a respective mean squared error.

[0094] In some implementations, the NL caption for the real image is automatically generated based on processing the real image using an additional model trained to predict captions for images.

[0095] In some implementations, generating the edited image using the text-guided diffusion model includes using the text-guided diffusion model without tuning any weights of the text-guided diffusion model.

[0096] In some implementations, the edit includes a replacement, of a subset of tokens of the source NL prompt, with one or more replacement tokens that differ from the subset of tokens of the source NL prompt.

[0097] In some implementations, the one or more edits include an addition, of one or more additional tokens, to the NL prompt.

[0098] In some implementations, the one or more edits include an adjustment of emphasis on one or more emphasis tokens of tokens of the NL prompt, the adjustment of emphasis being an increase or decrease of emphasis.

[0099] In some implementations, obtaining the edited NL caption includes obtaining the edited NL caption based on user interface input that edits a display of the NL caption for the real image. In some versions of those implementations, the user interface input includes typed input and/or an interaction with a graphical user interface that renders the display of the NL caption. In some of those versions, the edit includes an adjustment of emphasis on one or more emphasis tokens of tokens of the NL caption, the adjustment of emphasis being an increase or decrease of emphasis – and the user interface input includes the interaction with the graphical user interface and the interaction includes interaction with a slider that corresponds to the one or more emphasis tokens.

[00100] In some implementations, the user interface input includes spoken input that is captured in audio data and the method further includes processing the audio data, using an automatic speech recognition model, to generate recognized text that corresponds to the spoken input, and processing the recognized text to determine the edited NL caption.

[00101] In some implementations, a method implemented by processor(s) is provided and includes identifying a real image captured by a real camera and identifying a natural language (NL) caption for the real image. The method further includes performing, using a first guidance scale, a deterministic denoising diffusion implicit model (DDIM) inversion on the real image to generate a diffusion trajectory for the image. The method further includes, subsequent to generating the diffusion trajectory for the image, optimizing a sequence of unconditional embeddings, that are not based on the NL caption for the real image. Optimizing the sequence of unconditional embeddings can be based on using the diffusion trajectory for the image as a pivot during the optimizing and can be based on predicted latent codes generated during processing, using a text-guided diffusion model and a second guidance scale, that is based on a conditional embedding that is conditioned based on the NL caption for the real image. The method further includes using the optimized sequence of unconditional embeddings, and a

noise vector from the diffusion trajectory, in generating an edited image that is based on an edited NL caption that is an edited version of the NL caption for the real image. The edited image is visually similar to the real image but includes visual modifications consistent with one or more edits that are reflected by the edited NL caption.

[00102] These and other implementations of the technology disclosed herein can include one or more of the following features.

[00103] In some implementations, the first guidance scale causes the unconditional embeddings to have a lesser impact, in text-guided diffusion, than does the second guidance scale. For example, the first guidance scale can cause the unconditional embeddings to have no impact in text-guided diffusion.

[00104] Other implementations can include a non-transitory computer readable storage medium storing instructions executable by one or more processor(s) (e.g., a central processing unit(s) (CPU(s)), graphics processing unit(s) (GPU(s)), and/or tensor processing unit(s) (TPU(s))) to perform a method such as one or more of the methods described herein. Yet other implementations can include a system of one or more computers that include one or more processors operable to execute stored instructions to perform a method such as one or more of the methods described herein.

Claims

We Claim:

1. A method implemented by one or more processors, the method comprising:

identifying a real image captured by a real camera;

identifying a natural language (NL) caption for the real image;

performing an inversion on the real image to generate a diffusion trajectory for the real image, the diffusion trajectory including an encoding of the real image, a final noise vector, and a sequence of intermediate noise vectors between the encoding and the final noise vector;

subsequent to generating the diffusion trajectory for the real image:

optimizing a sequence of unconditional embeddings that are not based on the NL caption for the real image, wherein optimizing the sequence of unconditional embeddings is based on comparisons of generated predicted latent codes to the diffusion trajectory for the real image, each of the generated predicted latent codes being generated during a respective time step of processing, using a text-guided diffusion model, that is based on a conditional embedding that is conditioned based on the NL caption for the real image;

obtaining an edited NL caption that is an edited version of the NL caption for the real image;

generating an edited image that is visually similar to the real image but that includes visual modifications consistent with one or more edits that are reflected by the edited NL caption, wherein generating the edited image comprises generating the edited image based on processing, using the text-guided diffusion model:

the final noise vector,

the optimized sequence of unconditional embeddings, and

an edited conditional embedding that is conditioned based on the edited NL caption.

2. The method of claim 1,

wherein performing the inversion comprises using, during the inversion a first guidance scale, for classifier-free guidance of the text-guided diffusion model; and

wherein optimizing the sequence of unconditional embeddings comprises using, during processing using the text-guided diffusion model, a second guidance scale for classifier-free guidance of the text-guided diffusion model; and

wherein the first guidance scale causes the unconditional embeddings to have a lesser impact, in text-guided diffusion, than does the second guidance scale.

3. The method of claim 2, wherein the first guidance scale causes the unconditional embeddings to have no impact in text-guided diffusion.
4. The method of claim 2 or claim 3, wherein generating the edited image comprises using the second guidance scale during processing, using the text-guided diffusion model, to generate the edited image.
5. The method of any preceding claim, wherein generating the edited image based on processing, using the text-guided diffusion model: the final noise vector, the optimized sequence of unconditional embeddings, and the edited conditional embedding comprises:

using the edited conditional embedding in each of a plurality of time steps of the processing; and

using a corresponding next one of the unconditional embeddings in each of the plurality of time steps of the processing.

6. The method of any preceding claim, wherein performing the inversion on the real image comprises performing the inversion using a deterministic denoising diffusion implicit model (DDIM) inversion process.
7. The method of claim 6, wherein performing the inversion using the DDIM inversion process comprises performing DDIM sampling in reverse order and starting with the encoding of the real image.
8. The method of any preceding claim, wherein the comparisons of the generated predicted latent codes to the diffusion trajectory for the image each comprise a respective mean squared error.
9. The method of any preceding claim, wherein the NL caption for the real image is automatically generated based on processing the real image using an additional model trained to predict captions for images.

10. The method of any preceding claim, wherein generating the edited image using the text-guided diffusion model comprises using the text-guided diffusion model without tuning any weights of the text-guided diffusion model.
11. The method of any preceding claim, wherein the edit comprises a replacement, of a subset of tokens of the source NL prompt, with one or more replacement tokens that differ from the subset of tokens of the source NL prompt.
12. The method of any preceding claim, wherein the one or more edits comprise an addition, of one or more additional tokens, to the NL prompt.
13. The method of any preceding claim, wherein the one or more edits comprise an adjustment of emphasis on one or more emphasis tokens of tokens of the NL prompt, the adjustment of emphasis being an increase or decrease of emphasis.
14. The method of any preceding claim, wherein obtaining the edited NL caption comprises obtaining the edited NL caption based on user interface input that edits a display of the NL caption for the real image.
15. The method of claim 14, wherein the user interface input comprises typed input and/or an interaction with a graphical user interface that renders the display of the NL caption.
16. The method of claim 15,

wherein the edit comprises an adjustment of emphasis on one or more emphasis tokens of tokens of the NL caption, the adjustment of emphasis being an increase or decrease of emphasis; and

wherein the user interface input comprises the interaction with the graphical user interface and wherein the interaction comprises interaction with a slider that corresponds to the one or more emphasis tokens.

17. The method of any preceding claim, wherein the user interface input comprises spoken input that is captured in audio data and further comprising:

processing the audio data, using an automatic speech recognition model, to generate recognized text that corresponds to the spoken input; and

processing the recognized text to determine the edited NL caption.

18. A method implemented by one or more processors, the method comprising:

identifying a real image captured by a real camera;

identifying a natural language (NL) caption for the real image;

performing, using a first guidance scale, a deterministic denoising diffusion implicit model (DDIM) inversion on the real image to generate a diffusion trajectory for the image;

subsequent to generating the diffusion trajectory for the image:

optimizing a sequence of unconditional embeddings, that are not based on the NL caption for the real image, wherein optimizing the sequence of unconditional embeddings is based on using the diffusion trajectory for the image as a pivot during the optimizing and is based on predicted latent codes generated during processing, using a text-guided diffusion model and a second guidance scale, that is based on a conditional embedding that is conditioned based on the NL caption for the real image; and

using the optimized sequence of unconditional embeddings, and a noise vector from the diffusion trajectory, in generating an edited image that is based on an edited NL caption that is an edited version of the NL caption for the real image,

wherein the edited image is visually similar to the real image but includes visual modifications consistent with one or more edits that are reflected by the edited NL caption.

19. The method of claim 18, wherein the first guidance scale causes the unconditional embeddings to have a lesser impact, in text-guided diffusion, than does the second guidance scale.
20. The method of claim 19, wherein the first guidance scale causes the unconditional embeddings to have no impact in text-guided diffusion.
21. A system comprising memory storing instructions and one or more processors operable to execute the instructions to perform the method of any preceding claim.
22. One or more transitory or non-transitory computer readable media storing instructions that, when executed by one or more processors, cause performance of the method of any of claims 1 to 20.

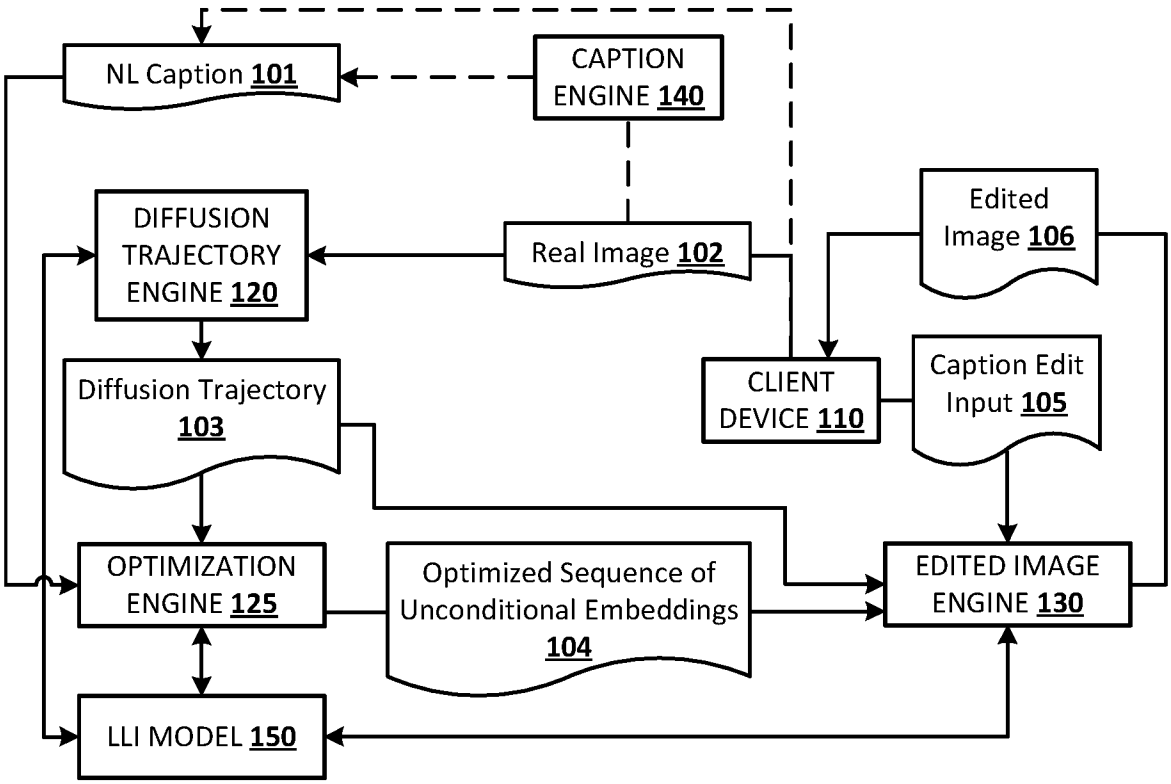

NL Caption 101
CAPTION ENGINE 140
DIFFUSION TRAJECTORY ENGINE 120
Real Image 102
Edited Image 106
Diffusion Trajectory 103
CLIENT DEVICE 110
Caption Edit Input 105
OPTIMIZATION ENGINE 125
Optimized Sequence of Unconditional Embeddings 104
EDITED IMAGE ENGINE 130
LLI MODEL 150

FIG. 1

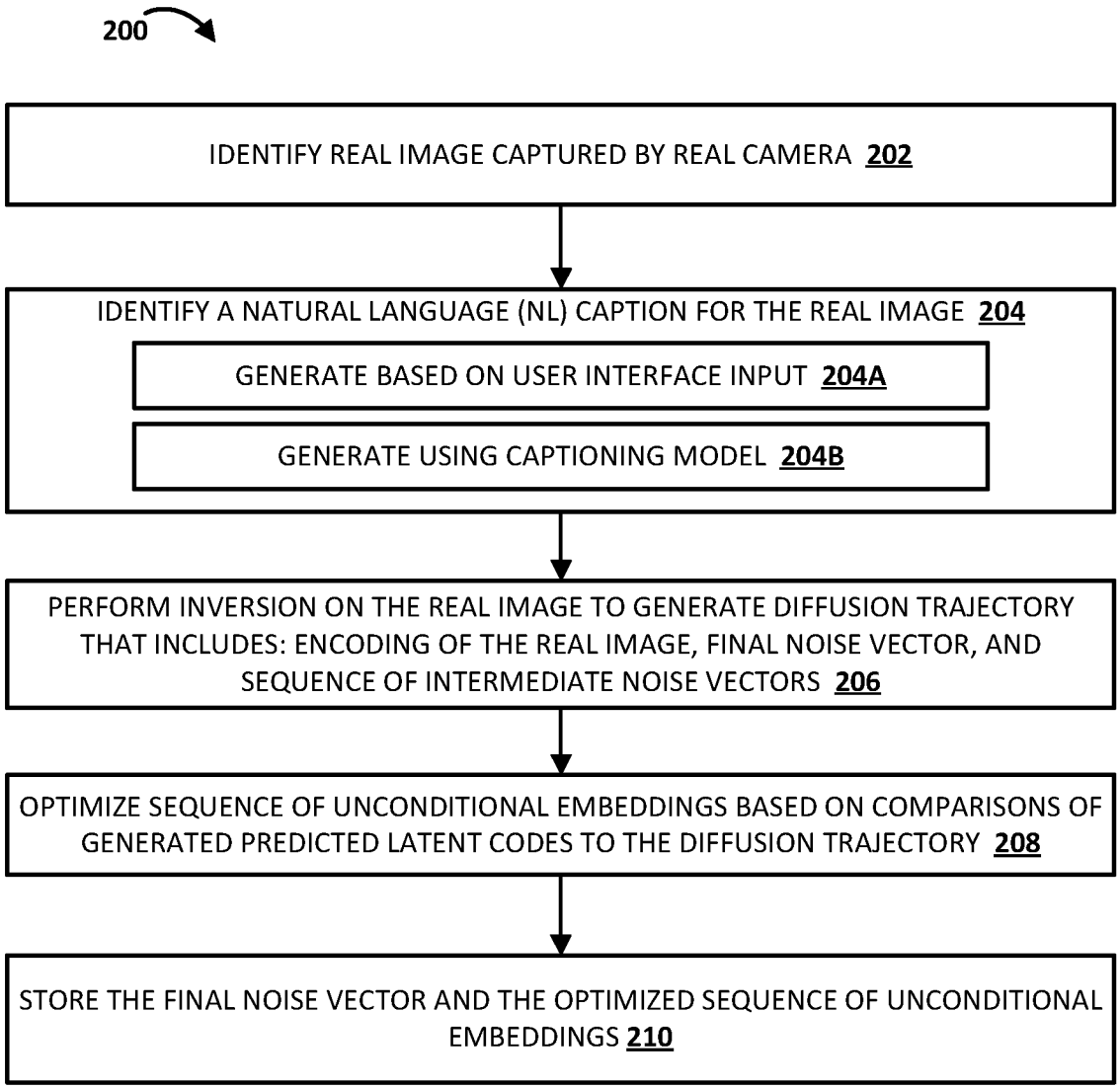

200
IDENTIFY REAL IMAGE CAPTURED BY REAL CAMERA 202
IDENTIFY A NATURAL LANGUAGE (NL) CAPTION FOR THE REAL IMAGE 204
GENERATE BASED ON USER INTERFACE INPUT 204A
GENERATE USING CAPTIONING MODEL 204B
PERFORM INVERSION ON THE REAL IMAGE TO GENERATE DIFFUSION TRAJECTORY THAT INCLUDES: ENCODING OF THE REAL IMAGE, FINAL NOISE VECTOR, AND SEQUENCE OF INTERMEDIATE NOISE VECTORS 206
OPTIMIZE SEQUENCE OF UNCONDITIONAL EMBEDDINGS BASED ON COMPARISONS OF GENERATED PREDICTED LATENT CODES TO THE DIFFUSION TRAJECTORY 208
STORE THE FINAL NOISE VECTOR AND THE OPTIMIZED SEQUENCE OF UNCONDITIONAL EMBEDDINGS 210

FIG. 2

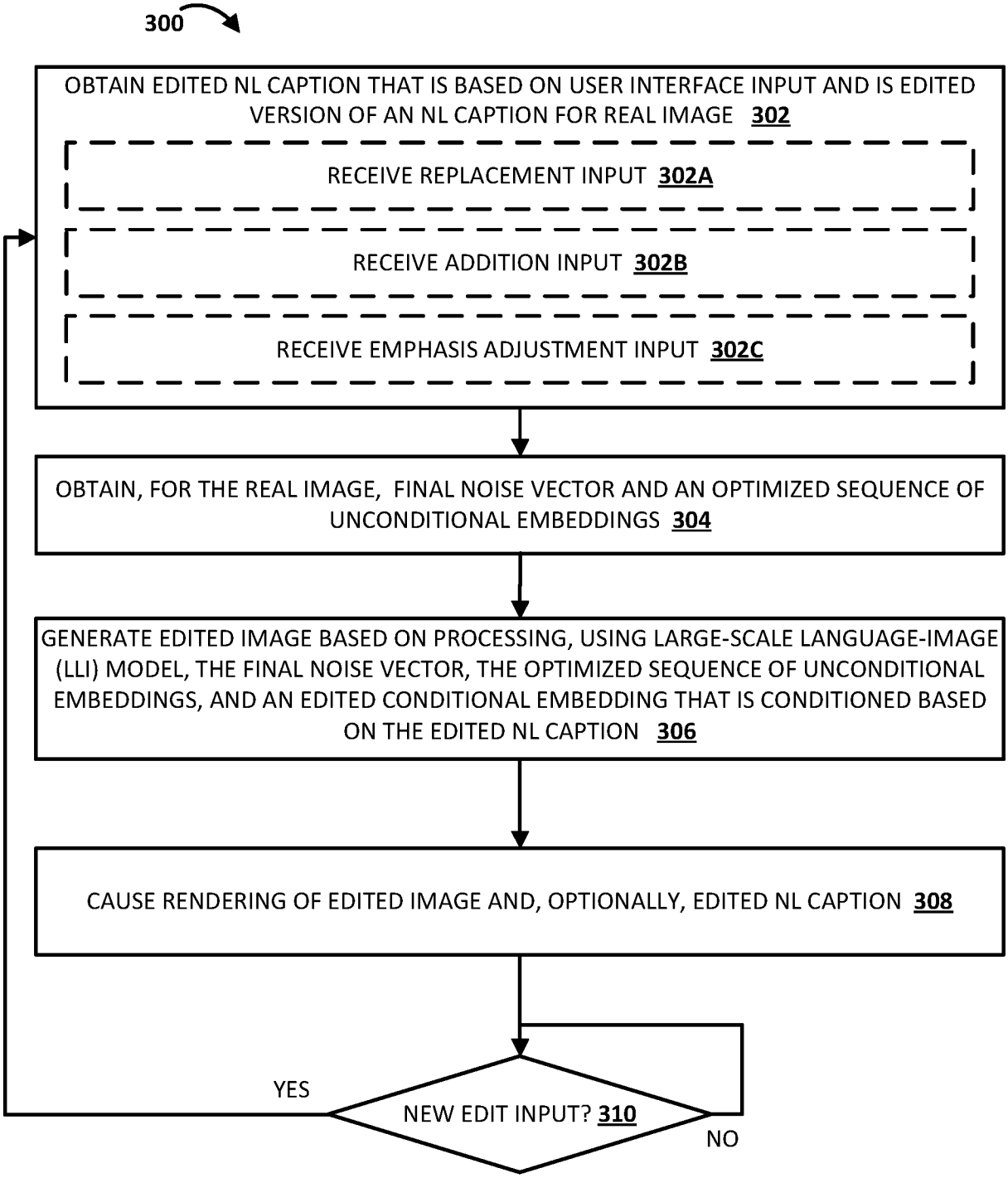

300
OBTAIN EDITED NL CAPTION THAT IS BASED ON USER INTERFACE INPUT AND IS EDITED VERSION OF AN NL CAPTION FOR REAL IMAGE 302
RECEIVE REPLACEMENT INPUT 302A
RECEIVE ADDITION INPUT 302B
RECEIVE EMPHASIS ADJUSTMENT INPUT 302C
OBTAIN, FOR THE REAL IMAGE, FINAL NOISE VECTOR AND AN OPTIMIZED SEQUENCE OF UNCONDITIONAL EMBEDDINGS 304
GENERATE EDITED IMAGE BASED ON PROCESSING, USING LARGE-SCALE LANGUAGE-IMAGE (LLI) MODEL, THE FINAL NOISE VECTOR, THE OPTIMIZED SEQUENCE OF UNCONDITIONAL EMBEDDINGS, AND AN EDITED CONDITIONAL EMBEDDING THAT IS CONDITIONED BASED ON THE EDITED NL CAPTION 306
CAUSE RENDERING OF EDITED IMAGE AND, OPTIONALLY, EDITED NL CAPTION 308
NEW EDIT INPUT? 310
YES
NO

FIG. 3

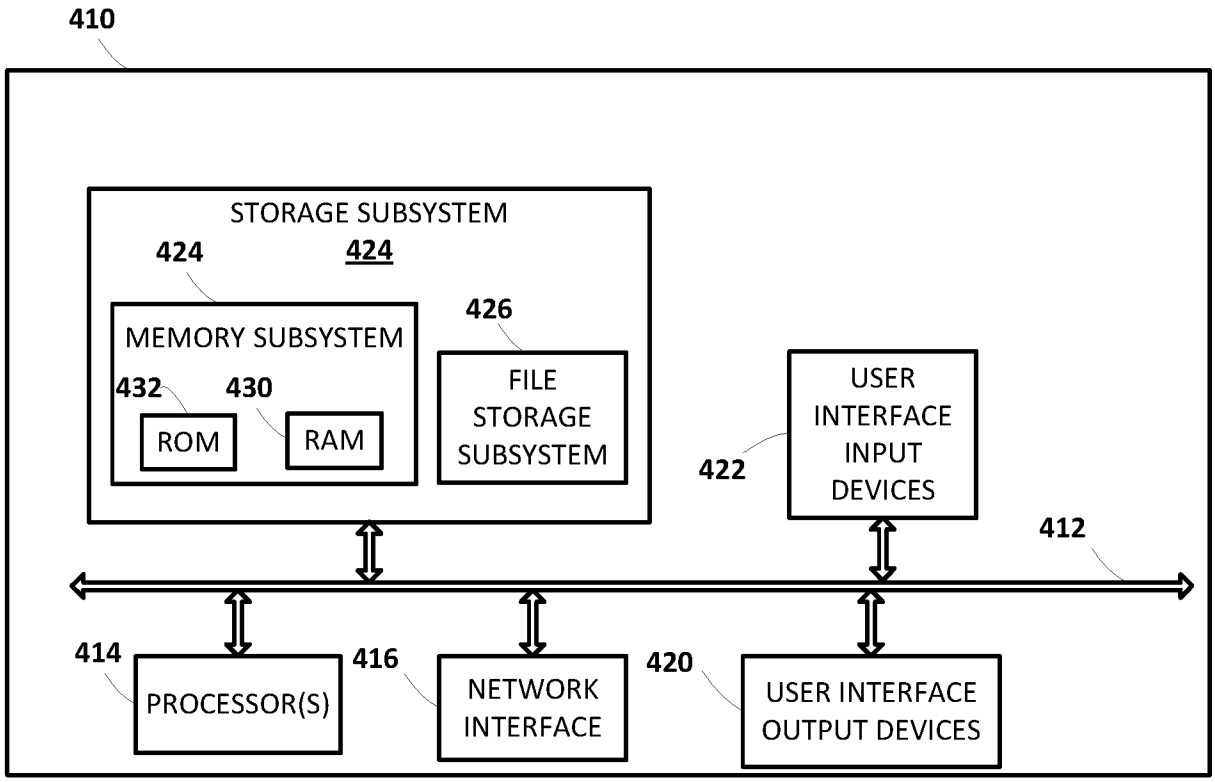

410
STORAGE SUBSYSTEM
424
424
426
MEMORY SUBSYSTEM
432
430
ROM
RAM
FILE STORAGE SUBSYSTEM
422
USER INTERFACE INPUT DEVICES
412
414
PROCESSOR(S)
416
NETWORK INTERFACE
420
USER INTERFACE OUTPUT DEVICES

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/079884

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06T11/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	AMIR HERTZ ET AL: "Prompt-to-Prompt Image Editing with Cross Attention Control", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 2 August 2022 (2022-08-02), XP091285831, abstract section 3 -----	1-22
A	GUILLAUME COUAIRON ET AL: "DiffEdit: Diffusion-based semantic image editing with mask guidance", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 20 October 2022 (2022-10-20), XP091349279, abstract; figures 1, 3, 4, 10, 11, 12 sections 3-4 -----	1-22

-/--

[x] Further documents are listed in the continuation of Box C.
[ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 March 2024	15/03/2024

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Katartzis, Antonios

2

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/079884

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BAHJAT KAWAR ET AL: "Imagic: Text-Based Real Image Editing with Diffusion Models", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 17 October 2022 (2022-10-17), XP091346475, abstract section 3 figures 1-4 -----	1-22
X,P	RON MOKADY ET AL: "Null-text Inversion for Editing Real Images using Guided Diffusion Models", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 17 November 2022 (2022-11-17), XP091371895, the whole document -----	1-22

2