

Geometry-preserved image editing

Taeun Kwon¹ and Junseok Kwon^{1,2} 

¹Graduate School of Artificial Intelligence, Chung-Ang University, Seoul, South Korea

²School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea

✉ E-mail: jskwon@cau.ac.kr

As generative models become more advanced and essential, interest in using these models for image editing is growing. Nevertheless, conventional image editing methods still have many problems and can alter the intrinsic properties (e.g. geometry) of an object during the editing process. In this article, a novel geometry-preserved image editing method is presented, where key points, keyframe, or canny edges are utilized as inputs for geometric constraints. A text prompt is fed into the inversion pipeline along with the input image and key points/keyframes/canny for editing. Then, new images are generated based on the editing pipeline of the denoising diffusion implicit model. Experiments demonstrate that the method outperforms its baselines.

Introduction: Many prominent models in image generation have been proposed and have made significant progress, including Stable Diffusion [1], DALL-E 2 [2], and GLIDE [3]. With the increasing development and interest in generative models, related markets, such as GPT-Turbo, are also growing. In image editing, diffusion models have been widely used, including the stable diffusion model, the denoising diffusion probabilistic models (DDPM) [4], and the denoising diffusion implicit models (DDIM) [5]. The stable diffusion model reflects the correlation between text and images through cross-attention, considering their relationship and weighting guide information accordingly.

Image editing has advanced further with zero-shot single-image editing, where a model utilizes a prompt to transform an image that it has never seen during the training stage. The prompt determines the direction of the image transformation and typically employs sound, images, text, and semi-tangible maps. Methods for generating and editing images based on visual cues such as edge detection, keypoints, semantic maps, and depth estimation images have received significant attention. Using visual guides enables detailed modifications in images and facilitates specifying the subject of the modification. Text-based conditions can be more ambiguous than image-based conditions, leading to a broader range of potential outcomes.

However, because the methods above allow for such a wide range of modifications, they inadvertently attempt to alter intrinsic properties in an image, which are not intended to change. For example, when editing images from a dog to a cat, conventional methods may also alter the geometric properties of the objects, such as transforming sideways objects into forward-facing ones (Figure 1). In addition, editing undesirable geometric properties induces erroneous geometric parts of the object. This occurs because conventional methods attempt to generate various images without adhering to geometric constraints on views or poses.



Fig. 1 Motivation: Conventional image editing methods often attempt to alter intrinsic properties, such as the geometric information of the object in an image, which are not intended to change

To address this problem, we present a geometry-preserved image editing method. Our proposed approach estimates the object's pose, extracts keyframes and keypoints, and utilizes this geometric information to generate a view-specific image within the DDIM framework. The contributions of our method are as follows.

- We propose a geometry-preserved image editing method based on the object's pose, keyframes, and keypoints, and impose geometric constraints on existing DDIM approaches. Notably, our constraint module can be seamlessly integrated into conventional diffusion models for image editing.
- We conducted a comparative analysis with other image editing models, including a Geometry-Preserved Comparison. Through this analysis, we demonstrated that our model, guided by geometric constraints, effectively transforms images while preserving the shape (size and orientation) of the input image and retaining the key features of the object.

Preliminary: DDIM [5] has evolved from the Markovian chain in DDPM [4] to a non-Markovian chain. Both DDPM and DDIM involve a forward process and a reverse process. In the forward process, noise is added to an image, while the reverse process uses UNet [6] to denoise the noisy image and restore the original image.

$$q_{\theta}(x_{1:T}|x_0) := q_{\theta}(x_T|x_0) \prod_{t=2}^T q_{\theta}(x_{t-1}|x_t, x_0), \quad (1)$$

where x_t denotes the latent variable in the sample space at time t and $q_{\theta}(x_{1:T}|x_0)$ describes a Markov chain with Gaussian transitions parameterized by θ , representing the inference distribution over the latent variables $x_{1:T}$. Unlike DDPM, DDIM operates as a non-Markovian chain, not relying solely on the previous step. Consequently, there is no need to proceed with all the sampling processes in order, resulting in increased sampling speed:

$$q_{\theta}(x_{t-1}|x_t, x_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 I\right), \quad (2)$$

where σ denotes the variance of the Gaussian distribution and α schedules the Gaussian transition.

Variational AutoEncoders (VAEs) [7] consists of two components: an encoder and a decoder. The VAE utilizes the encoder to extract a latent vector that effectively represents the input image. An input image is transformed into a small latent vector, resulting in a memory reduction. The decoder then uses this latent vector to generate an image that resembles the input while introducing new data. The VAE employed in this study is an AutoEncoder with a Kullback-Leibler (KL) divergence loss, which helps manage memory through tiling and slicing techniques. KL Divergence quantifies the difference between two probability distributions and is also utilized in DDIM to regularize noise predictions.

Geometry-aware image editing: Geometric constraint refers to imposing geometric restrictions on a subject during image editing. The proposed method utilizes keypoints, keyframe, canny as geometric constructions to perform view-specific or view-preserved image editing. For example, when using a keyframe as a constraint, the generated image is influenced by the shape of that keyframe. Our method can be implemented within the frameworks of both VAE and DDIM. Figure 2 illustrates the overall pipeline of the proposed method.

Geometry-aware image editing within the VAE: Input feature maps, derived from the original image through pose estimation [8], are transformed to generate keypoint heatmaps. These keypoint heatmaps visualize the importance of specific regions in the image, allowing us to extract keypoints and keyframes. The original image and the keypoints image are separately encoded into latent vectors using a Variational AutoEncoder, which are then fused together. During the fusion process and subsequent image editing, CrossAttention and text prompts guide the image changes according to the editing directions indicated by the prompt. Finally, the fused latent vector is decoded into an image.

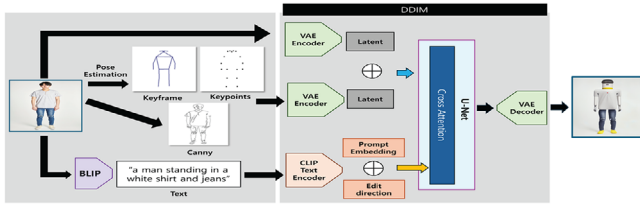


Fig. 2 Overall pipeline of the proposed method. The input image generates keypoints and keyframes through pose estimation. A text caption for the image is generated using BLIP. The input image is encoded using a VAE encoder to obtain a latent representation. The keypoints or keyframes are also encoded using the VAE encoder to obtain latent representations. These two latent representations are combined in a certain ratio to form a single latent representation, which is then subjected to DDIM inversion. During the cross-attention process with the noise image and prompt embedding, the latent representation is gradually transformed into one containing robot information. The VAE decoder then reconstructs the image, creating a new image

Geometry-aware image editing within the DDIM: The image is inverted and edited using DDIM. The VAE described above is integrated within the DDIM process. Before the DDIM inversion process, keypoints and keyframes are extracted, and the latent features are obtained using the VAE encoder as previously explained. Then, the concatenated latent features are inverted through multiple steps using a scheduler. This inversion process generates x_{inv} with the shape $[1, 4, 64, 64]$, and $x_{inv}[0]$ is saved. In the editing stage, the hidden latent is obtained by applying cross-attention between the inverted image and the prompt edit embedding, which is the text prompt embedding with the edit direction added. The edit direction is determined by the difference between the average embeddings of the sentences in person.txt and robot.txt. The embeddings of multiple images are averaged to obtain a single vector that captures the general characteristics of those images. The edited image is obtained by decoding with the VAE decoder and applying post-processing.

$$q_{\theta}(x_{1:T}|x_0 + x_{constrain}) := q_{\theta}(x_T|x_0 + x_{constrain}) \prod_{t=2}^T q_{\theta}(x_{t-1}|x_t, x_0 + x_{constrain}), \quad (3)$$

where this equation represents the inversion process of DDIM, defining the noise distribution from $x_0 + x_{constrain}$ to x_T . Keypoints or keyframe images are added to the original image x_0 using this formulation. Here, x_0 is the original image, and $x_{constrain}$ refers to the geometrically constrained image, such as keypoints, keyframes, or canny edges. x_t and x_{t-1} are the images at timestep t and $t - 1$, respectively.

Implementation details: The proposed method utilizes text and constrained images. First, we determine the desired transformation target. For example, if we aim to convert person images to robot images, we collect 1000 sentences describing images of people and 1000 sentences describing images of robots. Since ChatGPT generates almost identical sentences with different words when using the same prompt, we created 1000 sentences about humans and robots using various prompts. For example, to create person.txt, we used prompts like “Generate 200 sentences in detail about a standing person’s appearance, including arms, legs, feet etc.” To create robot.txt, we used prompts like “Imagine what a future robot will look like and generate 200 sentences.” Then, we obtain images for each category and resize them to RGB channel images with a size of 512×512 . Subsequently, we extract the keyframes and keypoints using pose estimation methods [8]. Keypoints represent points obtained through a heatmap in a circular format, while keyframes depict connections between keypoints. The canny images were created using OpenCV, with thresholds set to 100, 200 or 50, 150 to capture important shapes.

The image and text undergo an inversion process. Initially, a prompt is generated from the original image using the BLIP pretrained model. The inversion pipeline is based on the stable diffusion model. The image, keypoint or keyframe image, and text prompt are entered into the inversion pipeline. Subsequently, the original image and the keypoint or keyframe image pass through their respective VAEs to obtain latent rep-

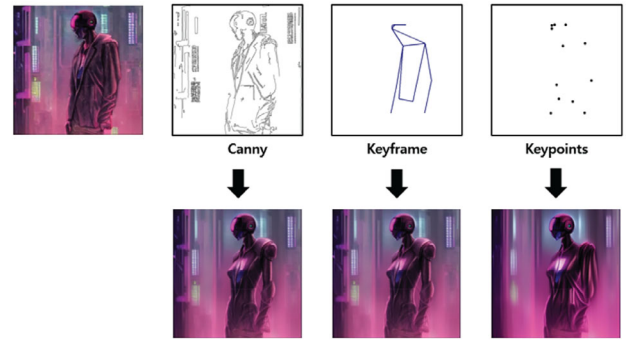


Fig. 3 Ablation study based on different geometric constraints



Fig. 4 Visual comparison with other models

resentations. The text prompt is processed through the text encoder of CLIP to create a prompt embedding. The latent vector from the original image is combined with the latent vector from keypoints/keyframes.

In the image editing step, the image is transformed using edit directions obtained from the previously created embeddings. Each edit direction is calculated based on the difference in the average embeddings. The base model for DDIM is stable diffusion, with the guidance amount and guidance scale set to 0.1 and 0.5, respectively. These parameters determine how much the image will be influenced by the prompt during image editing. The step t for the inversion and edit process of DDIM is 50. These modified prompt embeddings are then input into the UNet, and CrossAttention is performed on the inverted image and the prompt embedding edits. In CrossAttention, the inverted image serves as the Query, while the prompt embedding edits are used as the Key and Value.

Experiments: We used an NVIDIA 3090, CUDA 11.6, and Python 3.8.

Ablation study: We conducted an ablation study using different geometric constraints, such as Canny edges, keyframes, and keypoints of the original image, as shown in Figure 3. We utilized the original image from pix2pix [9]. In the experiment, the color of the Canny edge image was inverted to emphasize the effect on the edges. Figure 3 demonstrates that visual guides significantly influence the proposed method. For instance, when using the Canny constraint, the clothing of the object is detailed, which is reflected in the intricate patterns of the clothes in the image edited from person to robot. When using the keyframe constraint, the waist area of the object appears thinner compared to the Canny constraint, and the restored image also shows a thinner waist. Keypoints, representing significant points of the object, are derived from pose estimation and are visible when the image is enlarged. Unlike the keyframe constraint, if the keypoint constraint is used, the object’s arms are not connected, which is also reflected in the restored image. The experimental results indicate that the proposed method makes the best use of visual guides such as keyframes and keypoints.

Comparison with other models: As shown in Figure 4, We conducted comparative experiments with other image editing models. In the baseline model, Pix2Pix-Zero [10], pretrained models for dogs and cats are available, but not for people and robots, so the editing direction was set from dog to cat. Unnecessary background elements were excluded from

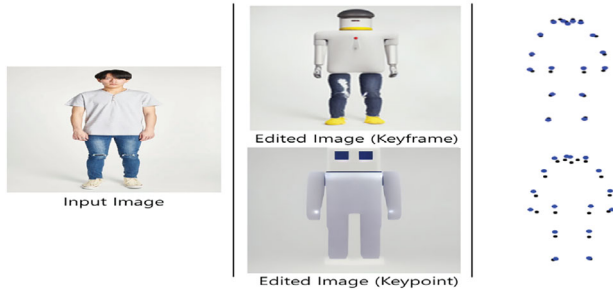


Fig. 5 Geometry-preserved comparison



Fig. 6 Zero-shot single image editing

Table 1. Quantitative comparisons in terms of LPIPS [14]

Constraints	Baseline [10]	Ours (with keypoints)	Ours (with keyframe)	Ours (with Canny)
LPIPS [14]	0.1469	0.2116	0.1311	0.1176

the input image, and a Canny edge image was obtained. In Pix2Pix-Zero [10], it was observed that the cat's head was slightly tilted downward. In ControlNet [11], both the color of the animal's fur and the background changed significantly. When using the prompt "cat" in ControlNet [11], the cat's face was turned to the right, but issues such as missing ears and overall distortion of the cat's form were observed. In Prompt-to-Prompt [12], while the dog was successfully transformed into a cat with its face turned to the right, the face was significantly distorted, and the essential feature of the cat's ears was not generated. In contrast, our model preserved the cat's overall form, including the ears, while successfully turning the cat's head to the right.

Geometry-preserved comparison: As shown in Figure 5, the image preservation performance was compared by analyzing the keypoints of the original and edited images. The keypoints of the original image are shown in black, while those of the edited image are in blue. In both cases, the image was edited from a human to a robot, with both images retaining the robot form. When geometry-preserved editing was performed using keypoints, it was observed that the humanoid form of the robot was created, reflecting the keypoints from the original image. The image edited through geometry-preserved editing using keyframes more accurately retained the original form compared to keypoint-based editing.

Qualitative and quantitative results: For this experiment, the proposed method learned image embeddings through cross-attention, incorporating the original image and text embeddings containing the edit direction. We utilized image embeddings from the original image using keyframes from pose estimation. After modifying the text prompt from "man" to "woman" and adding the words "with long hair," image editing was performed. Upon examining the image on the right in Figure 6, it is evident that it has been edited to depict a humanoid robot face with a grayish tint and long hair. This result demonstrates the effective integration of the text prompt with the image fused with keyframes.

Table 1 compares the experimental results in terms of perceptual image patch similarity (LPIPS) on the person jogging dataset. LPIPS evaluates the semantic similarity between two images by computing the similarity between the activations of two image patches for a pre-defined model. For this experiment, we used SqueezeNet as the pre-defined model, which has a small number of parameters and low capacity requirements. A low LPIPS value indicates that the image patches are

semantically similar. Table 1 shows that the LPIPS values of images including keypoints or keyframes decrease compared to the baseline model, with values of 0.1016, 0.093, and 0.0966.

Computational cost: The execution time was measured using a single NVIDIA RTX 3090. The input image used was that of an Asian man, as shown in Figure 5. The time taken to generate keypoints and keyframes was 0.0754 s. The inversion process took 88.238 s, or approximately 1 min and 20 s. The editing process took 465.987 s, roughly 7 min and 40 s.

Conclusion and discussion: In this paper, we explored the capabilities of constraints within image editing. We demonstrated a decrease in the learned perceptual image patch similarity (LPIPS) value, indicating that the characteristics of the image were well preserved during the extraction and fusion of latent vectors through AutoEncoderKL. Furthermore, we observed a reduction in significant changes to the overall skeleton or the appearance of artifacts, such as spurious hands, in the edited images. Additionally, it was evident that detailed elements, such as clothing patterns, were preserved rather than distorted. We consider this study to be effective and plan to further investigate its applications in the future.

Our method can be extended for sound-guided semantic image manipulation, as described in [13]. For this, the latent vector from the sound can be obtained through an encoder and combined with the image to alter a person's facial expression. Our method can then be extended by incorporating text, images, and sound. Since sound can convey emotions and speech nuances, integrating it into the process is expected to enable the expression of more diverse emotions in images. For instance, when observing a scene near a shopping mall, specifying the viewpoint, such as looking from a certain angle or towards the left, can enhance the realism of the generated images. When transforming one object into another, it is ideal to accurately reflect only the desired characteristics of the object while enhancing aspects like background blur. It is essential to develop a versatile, high-performance model that integrates not only text and images but also sound for image transformation.

Author contributions: **Taeun Kwon:** Conceptualization; writing—original draft. **Junseok Kwon:** Conceptualization; writing—original draft.

Acknowledgements: This work was partly supported by the Chung-Ang University Research Grants in 2022, and partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-01341, Artificial Intelligence Graduate School Program (Chung-Ang university)).

Conflict of interest statement: The authors declare no conflicts of interest.

Data availability statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

© 2024 The Author(s). *Electronics Letters* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Received: 30 June 2024 Accepted: 19 August 2024
doi: 10.1049/ell2.70011

References

- Rombach, R., et al.: High-resolution image synthesis with latent diffusion models. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway (2022)
- Ramesh, A., et al.: Hierarchical text-conditional image generation with clip latents. arXiv preprint, arxiv:2204.06125 (2022)
- Nichol, A.Q., et al.: GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In: ICML. International Machine Learning Society, Madison, WI (2022)

- 4 Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. arXiv preprint, arxiv:2006.11239 (2020)
- 5 Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR. IEEE Information Theory Society, Piscataway (2021)
- 6 Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. Springer, Cham (2015)
- 7 Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint, arXiv:1312.6114 (2013)
- 8 Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: ECCV. Springer, Berlin (2018)
- 9 Isola, P., et al.: Image-to-image translation with conditional adversarial networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway (2017)
- 10 Parmar, G., et al.: Zero-shot image-to-image translation. In: ACM SIGGRAPH. ACM, New York (2023)
- 11 Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847. IEEE, Piscataway (2023)
- 12 Hertz, A., et al.: Prompt-to-prompt image editing with cross attention control (2022). <https://arxiv.org/abs/2208.01626>
- 13 Lee, S.H., et al.: Sound-guided semantic image manipulation. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway (2022)
- 14 Zhang, R., et al.: The unreasonable effectiveness of deep features as a perceptual metric. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway (2018)