Casper DPM - Supplementary Materials

Anonymous Author(s)

1 IMPLEMENTATION DETAILS

1.1 Baseline Pipeline

1.1.1 Calibration. For completeness we mention different failed attempts to calibrate the LMC and the camera: we tried to directly triangulate retro reflectors or IR point light sources using the LMC raw images, and using hand-like objects with known geometry to imitate the human hand [Guna et al. 2014]. Notably, despite previous studies reporting good accuracies in calibrating the LMC with or without other optical devices [Moser and Swan 2016a,b; Weichert et al. 2013], we did not manage to achieve such results. The reported joint locations from the LMC API were observed to be noisy even in the recommended working distance and our final calibration yielded errors of approximately 5mm in the peripheral area of the LMC.

1.1.2 Skinning. The mesh we used throughout the paper is a modified quad-dominant mesh of the LMC open-source repository, and is not necessarily anatomically correct. The vertex weights per bone were normalized and the bone associated with the "elbow" joint was removed, as we observed the LMC predictions for the elbow are significantly error prone.

1.2 MLS

For the MLS Deformation step, we minimize a similarity transformation rather than rigid or affine transformations, as it worked best in practice. We used a discrete grid of size 20x20, spaced evenly per axis. We used an alpha value of 0.5 from the original paper [Schaefer et al. 2006], which dictates the deformation effect strength of a point as a function of distance from a control point. To save computation time of solving the grid (which happens every frame), we did not include the proximal interphalangeal joints for this operation, as they did not effect the deformation in a meaningful way.

1.3 Temporal Filtering

For an illustration of the different approaches for temporal filtering described in the paper, please see Figure 1.

1.4 JND User Study

We provide further detail about the modifications made to the experimental protocol proposed by Peng et al. [2023].

As mentioned before, we do not prerecord videos before (or during) the experiments. Instead, we synthesize the videos on-the-fly using the same graphics engine used by the live system, capable of roughly 920 FPS throughput. This has the large benefit of not needing to pre-render offline frames for subjects, which would otherwise be a requirement since the simulated latencies are adjusted according to their decisions. Furthermore, because we use the same rendering technique and projector for both the user study and the live system, this decreases the gap between the experimental conditions and when an actual hand is presented. On top of this, the high speed projector reduces blur artifacts, which would occur if a regular screen or projector were used. Secondly, we set the starting simulated latency to 20ms instead of 40ms, which is closer to the expected JND in these scenarios, and reduces overall subject trials. Finally, we did not choose to use two interleaved sessions (and averaging their result), as proposed originally to remove some bias from the experimental results. Instead, we chose to have more subjects participate and more analysis axis to analyze.

Additionally, prior to conducting the full user study, a small pilot with 3 subjects was held to validate our experimental setup and determine some fixed hyper parameters for the full user study which we estimated to have strong effects on the result. This included the overall simulation speed, the type of projected pattern and the intensity of the ambient lighting present in the simulated scene. The projector was simulated using conventional projective textures [Everitt 2001]. Interestingly, we found the projected pattern to have little to no effect on JND for the 3 subjects in the pilot.

To compute the heatmaps (main paper, Figure 10) out of subjects' marking, we first find the red disconnected components in the painting, and calculate their convex hull. Then, a pixel's final color in the heatmap is the weighted sum of the paintings from all users, where the weights are the reciprocal of the number of pixels each subject painted red. The result is interpolated for pixels missing a value, and pixels outside of the hand area are masked out. See Figure 2 for a representative image taken during the user study.

1.5 Guess the Character

See Figure 3 for a reference image showing an example from the user study.

1.6 Projegraphy

A schematic overview of the projegraphy pipeline is shown in Figure 4. GPT-4 [Achiam et al. 2023] is utilized initially to identify the animal, as our research revealed that employing ControlNet [Zhang et al. 2023] directly with a generic prompt and without a specific animal name (e.g., "A cute animal"), yields unsatisfactory visual results. Hyper-parameters for both GPT-4 and ControlNet were chosen by manually examining the results of a search-grid method on what we considered the main hyper-parameters. We used the implementation of ControlNet provided by Mikubill [2023]. The searched hyper-parameters for ControlNet, with their final values (in parenthesis - secondary less-used configuration), are as follows - stable diffusion model: "RealVisXL V3.0 Turbo" [SG161222 2023b] ("DreamShaper V8" [Lykon 2023]), prompt: "National Geographic Wildlife photo of <GPT-4 identified animal>" ("a cute <GPT-4 identified animal>"), steps: 7 (20), cfg scale: 1.5 (7), width: 512, height: 512, sampler name: "DPM++ SDE" ("DPM++ 2M"), ControlNet model: "diffusers_xl_canny_mid" [lllyasviel 2023], ControlNet weight: 2 (1.5), ControlNet guidance end: 1 (0.3). Other configuration parameters were set to default. Interestingly, the search-grid process

,
,
.

Table 1: System profiling breakdown.

Step	Time [ms]
Camera Acquisition	1.89*
Camera Frame Upload	0.65 ± 0.22
Skinning	0.34 ± 0.12
MLS Deformation	0.27 ± 0.051
PBR	0.43 ± 0.067
Download Render	1.5 ± 0.23
Projector Latency	3.0*
LMC Acquisition	9.1 ± 0.4
MLS Thread	16.2 ± 1.9

revealed that conditioning with Canny edge image yielded better results than other modalities such as depth, open-pose, and normals and that the stable diffusion model "DreamShaper V8" [Lykon 2023] outperforms other models such as "RealVisXL V3.0" [SG161222 2023a], "SDXL Turbo V1.0 fp16" [stabilityai 2023] and "MeinaMix V11" [meina 2023]. As the input mask doesn't adhere to the 1:1 ratio mandated by the stable diffusion model, and given that the hand-bounding box doesn't necessarily occupy the majority of the image, the input mask undergoes crop and resize before inference. Subsequently, the output image undergoes the reverse process, to match the input mask dimensions and ratio. For GPT-4, after experimenting with a few prompts, we found that the prompt that yielded the most reasonable and stable results was "This is a picture of a hand gesture. Which animal is it most similar to? Return 3 animals by priority in a JSON format (with no explanations).".

2 PROFILING

Our implementation running the full pipeline was profiled upon 5000 frames using NVIDIA Nsight Systems to get a rough idea of system latency. Results can be seen in Table 1. Entries with an asterisk (*) are estimated based on hardware specifications given our settings. The upper part of the table consists of entries directly effecting the end-to-end latency (i.e. their sum is a rough estimate of system latency). The bottom part contains information about operations that happen in parallel.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

Cass Everitt. 2001. Projective texture mapping. *White paper, NVidia Corporation* 4, 3 (2001).

Jože Guna, Grega Jakus, Matevž Pogačnik, Sašo Tomažič, and Jaka Sodnik. 2014. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors* 14, 2 (2014), 3702–3720.

lllyasviel. 2023. "DiffusersXL Canny Mid". https://huggingface.co/lllyasviel/sd_control_collection/resolve/main/diffusers_xl_canny_mid.safetensors. Accessed: 2024-03-20.

Lykon. 2023. "DreamShaper V8". https://civitai.com/models/4384/dreamshaper. Published: 2023-07-29.

meina. 2023. "MeinaMix V11". https://civitai.com/models/7240/meinamix. Published: 2023-07-16.

Mikubill. 2023. "sd-webui-controlnet". https://github.com/Mikubill/sd-webui-controlnet. Accessed: 2024-03-20.

Kenneth R Moser and J Edward Swan. 2016a. Evaluation of hand and stylus based calibration for optical see-through head-mounted displays using leap motion. In *2016 IEEE Virtual Reality (VR)*. IEEE, 233–234.

Kenneth R Moser and J Edward Swan. 2016b. Evaluation of user-centric optical see-through head-mounted display calibration using a leap motion controller. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 159–167.

Hao-Lun Peng, Shin'Ya Nishida, and Yoshihiro Watanabe. 2023. Studying user perceptible misalignment in simulated dynamic facial projection mapping. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 493–502.

Scott Schaefer, Travis McPhail, and Joe Warren. 2006. Image deformation using moving least squares. In *ACM SIGGRAPH 2006 Papers*. 533–540.

SG161222. 2023a. "RealVisXL V3.0". https://huggingface.co/SG161222/RealVisXL_V3.0/resolve/main/RealVisXL_V3.0.safetensors. Accessed: 2024-03-20.

SG161222. 2023b. "RealVisXL V3.0 Turbo". https://huggingface.co/SG161222/RealVisXL_V3.0_Turbo/resolve/main/RealVisXL_V3.0_Turbo.safetensors. Accessed: 2024-03-20.

stabilityai. 2023. "SDXL Turbo V1.0 fp16". https://huggingface.co/stabilityai/sdxl-turbo/resolve/main/sd_xl_turbo_1.0_fp16.safetensors. Accessed: 2024-03-20.

Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the accuracy and robustness of the leap motion controller. *Sensors* 13, 5 (2013), 6380–6393.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

Figure 3: Guess the Character User Study. An example round from the user study. A character is projected onto the hand and users must bend the appropriate finger. The characters are randomly moved around the finger tips so they may appear out of sight for a short while.

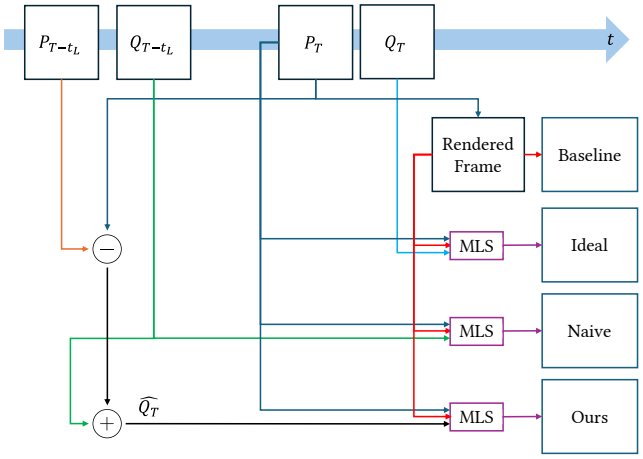

Figure 1: Temporal Filtering. Consider the current time T. The *baseline* solution involves using the rendered image created from P_T (the LMC frame) directly. If the tracker was instant, we would use Q_T as target points for P_T to deform the render giving us the *ideal* result. However the tracker is not instant, and *naïve*ly using the available Q_{T-t_L} would lead to lag and jitter. On the other hand, *our* filter takes into account velocity information from previous Ps, and estimates a relatively accurate $\widehat{Q_T} \approx Q_T$.

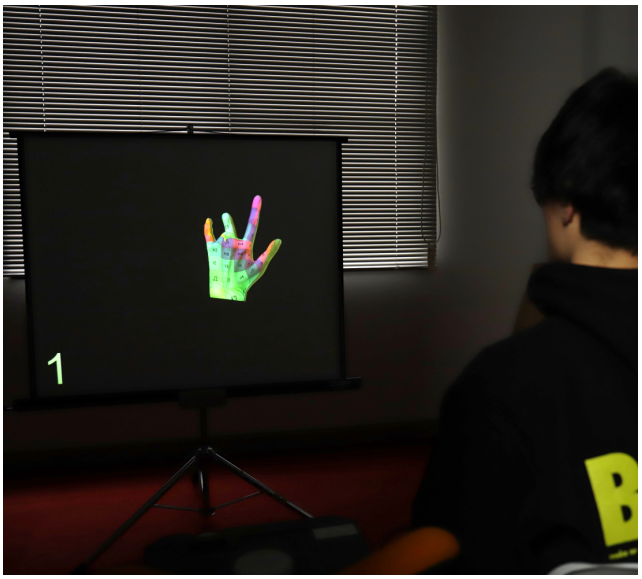

Figure 2: JND User Study. Subjects observed pairs of videos projected on a diffuse screen that were rendered live using our graphics engine.

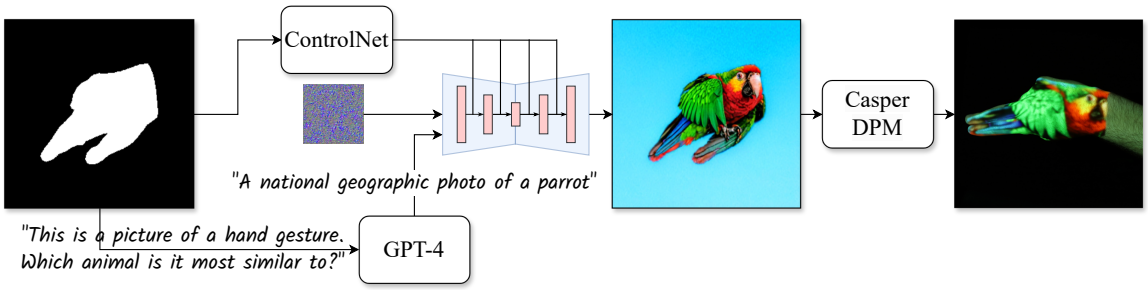

Figure 4: Projegraphy Pipeline. This pipeline generates and projects relevant live content according to the hand gestures, without any additional user input.