Good Keyframes to Inpaint

Shohei Mori, *Member, IEEE,* Dieter Schmalstieg, *Fellow, IEEE,* and Denis Kalkofen, *Member, IEEE*

Abstract—Diminished Reality (DR) propagates pixels from a keyframe to subsequent frames for real-time inpainting. Keyframe selection has a significant impact on the inpainting quality, but untrained users struggle to identify good keyframes. Automatic selection is not straightforward either, since no previous work has formalized or verified what determines a good keyframe. We propose a novel metric to select *good keyframes to inpaint*. We examine the heuristics adopted in existing DR inpainting approaches and derive multiple simple criteria measurable from SLAM. To combine these criteria, we empirically analyze their effect on the quality using a novel representative test dataset. Our results demonstrate that the combined metric selects RGBD keyframes leading to high-quality inpainting results more often than a baseline approach in both color and depth domains. Also, we confirmed that our approach has a better ranking ability of distinguishing good and bad keyframes. Compared to random selections, our metric selects keyframes that would lead to higher-quality and more stably converging inpainting results. We present three DR examples, automatic keyframe selection, user navigation, and marker hiding.

Index Terms—Diminished reality, inpainting, keyframe, good keyframes to inpaint, SLAM.

1 Introduction

Diminished reality (DR) removes physical objects from the user's view by estimating the color and geometry of the background [1]. DR can be beneficial in a variety of use cases, from reducing visual pollution by hiding markers [2] and scene instrumentation [3] to X-ray visualization by rearranging objects in the scene [4].

Two approaches can be distinguished to remove a region of interest (ROI). First, if the background behind the ROI can be directly observed, DR can make use of real background pixels. This approach is only feasible if it has access to recorded or streamed images where the background is exposed [3], [5], [6]. Otherwise, pixels close to the ROI in the current frame or a *keyframe* observed in the past must be used. This approach has no special prerequisites, but it requires real-time image inpainting [7], [8], [9].

The quality of the final result can only be as good as the keyframes allow, since it initiates the subsequent inpainting. However, not all keyframes are equally suitable. For example, good keyframes contain sufficient pixels with a high similarity to the background, which is usually assumed to resemble the surrounding of the ROI. Consequently, inpainting algorithms generate better results on images where the ROI is small, so that enough of the scene is visible.

Existing DR methods relying on keyframes depend on the user to select a set of suitable frames. However, this assumption can introduce a substantial bias. The resulting quality either depends on the user's experience in identifying good keyframes or on the time spent on adjusting the selection until a sufficient result has been generated.

We address this problem by introducing an approach for the automatic selection of keyframes. Thus, we introduce a metric for determining *good keyframes for inpainting*; a phrase that was inspired by seminal work in feature detection [10]. We identify multiple criteria that have an impact on the inpainting result, such as the image size to ROI ratio, and the pixel similarity in the vicinity of the ROI. We further introduce a set of weights to balance the impact of each individual criteria to the final score of a keyframe. We find the weighting based on an evaluation of the corresponding inpainting quality in a dataset that we generated from capturing real environments. More specifically, we determine the weights from combinations between the parameters and the inpainting quality, as observed in a collection of 360 image pairs.

The resulting metric enables the automatic selection of keyframes among the images encountered along a recorded camera path. To further improve the inpainting result, we provide user guidance towards suitable keyframes which is based on the proposed metric.

Fig. 1 shows two example keyframes in a scene and their *good keyframes to inpaint* scores together with the inpainting results using the keyframes. The highest scoring keyframe can be expected to yield high quality inpainting, as demonstrated in the corresponding inpainting result, while the lowest scoring keyframe results in a low quality inpainting result (e.g., the metal support is unnecessarily repeated, and the seams, especially at the rear, are disconnected). The proposed scoring enables the system to select good keyframes automatically, or, otherwise, it guides the user to a good keyframe. While Fig. 1 demonstrates RGBD inpainting, the proposed scoring can be applied to planar inpainting commonly used in existing DR systems [11].

In summary, our work significantly contributes to the state of the art in DR. Specifically, it presents the following three major contributions:

- We introduce a metric for estimating the suitability of a keyframe for inpainting a certain ROI.
- We introduce an importance ranking of our criteria, and we determine how to best balance evaluation metrics.
- We demonstrate the impact of the proposed approach on real application cases.

• *S. Mori, D. Schmalstieg, and D. Kalkofen are with the Institute of Computer Graphics and Vision, Graz University of Technology, Austria. E-mail: see https://www.icg.tugraz.at/*

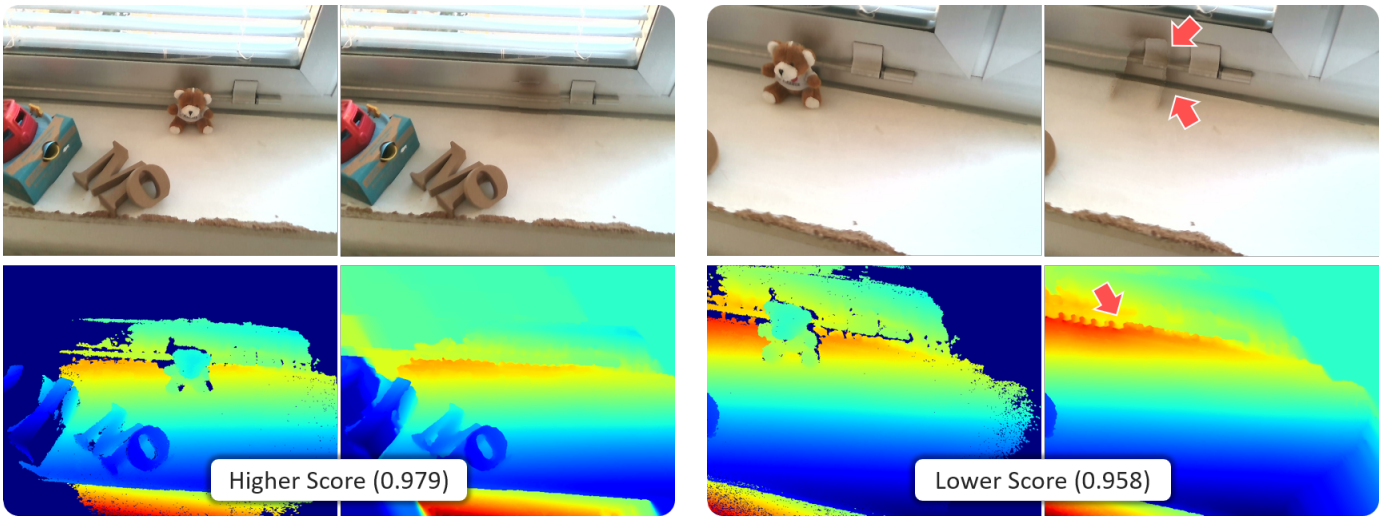


Fig. 1. The proposed *good keyframes to inpaint* method scores keyframe candidates. (left) A keyframe with a high score can be expected to yield inpainting with high quality in both color (top) and depth (bottom) domains. (right) A keyframe with a low score may end up with low quality inpainting. With the proposed scoring, the DR system is able to navigate the user to a good keyframe or select it automatically.

2 Related Work

DR can be categorized into approaches using multiple views and approaches relying on inpainting. Multi-view DR builds on 3D vision [5], [12] and image-based rendering techniques [3] to reconstruct the background occluded by the ROI object via information contained in the 3D reconstruction obtained from many other camera views [13]. While real-time video streams, for example, from surveillance cameras, can serve a similar purpose without requiring an offline reconstruction step [6], the number of live cameras is typically very limited in practice. Another source for views of the background are large public image databases [14].

DR inpainting fills in pixels within the ROI with only a single live camera as the image sources. Pixels in the vicinity of the ROI are rearranged to cover the ROI [8], [9], [15]. Our method falls into this latter category of DR.

A key insight of DR inpainting is prudent use of keyframes. Since the search for matching pixels to cover the ROI is costly, previous real-time approaches run inpainting only once on a keyframe (typically, in a background thread amortized over several frames) and warp the keyframe to the next frames. The warped keyframe is used to avoid repeated bootstrapping of the inpainting process and to preserve the appearance over multiple frame. Therefore, the first keyframe inpainting determines the overall quality.

A popular implementation trick for keyframe inpainting is to identify a dominant background plane and resample the keyframe to rectify this plane, thereby exposing candidate pixel clusters for inpainting with reduced perspective distortion. Such automatic background rectification relieves the user of having to choose a keyframe where the background is oriented in parallel to the image plane, but it can only be applied in near-planar scenes [9], [16], [17] and for known foreground object sizes [7], [18], [19]. In general scenes, rectifications cannot be applied [8], [15], making keyframe selection much more brittle. Our approach avoids these drawbacks and directly indicates to the user if a keyframe candidate is suitable for inpainting.

Alas, results of inpainting algorithms can be frustratingly unpredictable. One reason lies in the choice of random seeds used to bootstrap the inpainting algorithm [8], [20]. Since brute-force pixel searching of all pixels in the ROI is infeasible, practical solutions can provide, at best, local minima of plausible pixel combinations resulting from searching near the random seeds. Although it is hardly mentioned in the literature, previous inpainting methods commonly rely on the user to select good results from a gallery [21] or, otherwise, repeat the process. Such a user-in-the-loop workflow may be acceptable for image editing applications, but hardly for "immersive" DR experiences. In contrast, the objective of our method is to increase chances to deliver high-quality results in a manner that is largely independent of the choice of random seeds by making sure the keyframe used as source is sufficient even if only heuristic searching is feasible.

Finally, it is worth noting that all recent DR inpainting systems rely on exemplar-based inpainting. Recent neural network-based inpainting has not yet been applied to DR, likely, because the learning is highly dependent on the used dataset and prone to overfitting. One possible approach to mitigate such issues is to use observed neural features within the field of view (FOV) [22], [23], which can be interpreted as being inspired by exemplar-based inpainting. Because of this similarly, we expect that our metric may also be applicable to neural network methods. Having usable pixels within the FOV is a fundamental requirement for all inpainting methods, independent of how pixels are chosen.

3 Guidelines for Keyframe Selection

This section gives an overview of our method, which focuses on a single keyframe [2], [8], [9], [16], [19], [24]; an extension to multiple keyframes [15] is discussed in the outlook. Our metric for measuring the suitability of a keyframe combines several criteria. We start with heuristic guidelines for keyframe selection, from which we derive mathematical models describing criteria of our metric. We

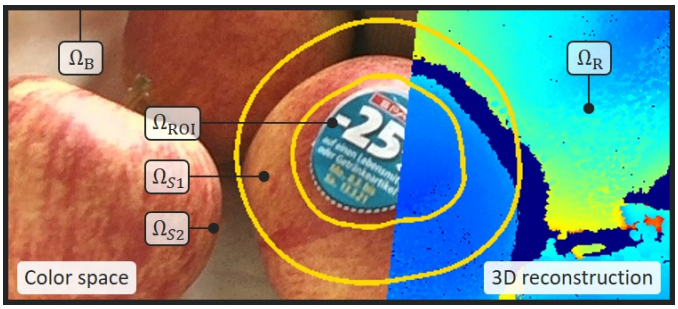


Fig. 2. Regions to define the keyframe suitability criteria. Within the FOV, we distinguish ROI and its surrounding areas as Ω_{ROI} and Ω_{S}, respectively. Further, we separate Ω_{S} into Ω_{S1} and Ω_{S2}, which denote boundary pixels of ROI with a margin, and the rest, respectively. Ω_{B} is a margin surrounding the FOV. The colored points represent areas Ω_{R} where scene reconstruction is completed.

Fig. 3. "Keep ROI fully in sight" (rule 1). The top and bottom rows show good and bad keyframes according to rule 1, respectively. The yellow lines in the left images indicate the ROI. With the bad keyframe, the algorithm propagates the wall pixels, invading the sink.

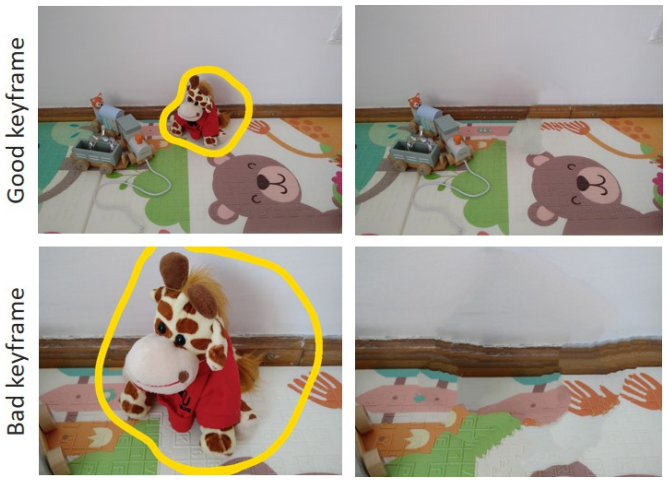


Fig. 4. "Secure source areas" (rule 2). The top and bottom rows show good and bad keyframes according to rule 2. In the bad keyframe, the algorithm distorts the straight lines on the wall. As the ROI is too large, the source pixel connections are lost repeatedly during the optimization, and new pixel candidates are found, which distort the lines.

conclude this section with the description of the weights required to adjust the impact of the criteria.

In addition to the six rules, which all specify geometric criteria, we also require photometric consistency. For example, motion blur due to a fast camera motion should be avoided, since it is baked into a keyframe and would be used indiscriminately for images that may or may not come from a fast moving camera. However, we do not discuss this further, since solutions can easily be embedded in the capturing process itself. For example, the user may be asked to stop for taking a keyframe, or a keyframe may be ignored when fast motion occurs.

Our method is on the basis of the state-of-the-art DR inpainting method for generic 3D scenes [15]. As such, we assume access to RGBD frames, a 3D bounding object, and surfels reconstructed by the internal SLAM system [25]. Fig. 2 depicts the regions we use. We define regions in a frame to calculate the proposed criteria as follows: Given a ROI, Ω_{ROI}, within the FOV, the inpainting algorithm fills the ROI using the pixels from the region surrounding the ROI, Ω_{S}. We separate Ω_{S} into two regions to distinguish close pixels to the ROI, Ω_{S1}, and the others, Ω_{S2} (i.e., $\Omega_{\mathrm{S}} = \Omega_{\mathrm{S1}} \cup \Omega_{\mathrm{S2}}$). Also, Ω_{B} defines a margin surrounding the FOV. Ω_{R} defines pixel areas of the projected SLAM surfels.

3.1 Keep ROI fully in sight (rule 1)

The ROI must be within the FOV, so we can completely inpaint all pixels of the target region. If a part of the ROI is clipped by the view frustum of the keyframe, no inpainting can be prepared for the clipped portion, leaving it uninitialized in the final image. Fig. 3 shows two inpainted keyframes; one successfully captures the entire ROI within the FOV, while the other fails to do so. The former can use pixels near the ROI in every direction, while the latter loses pixels on the bottom left, resulting in the pixel leaking from the wall to the washstand. Even if the ROI is not clipped, pixels of the ROI located near the image borders are more difficult to inpaint, since there are fewer potential source pixels in the vicinity.

Therefore, the first rule, *keep ROI fully in sight*, is considered mandatory. To enforce the rule, we define $c_{\mathrm{B}} \in \{0, 1\}$, a predicate that invalidates other criteria when it is 0. We detect if ROI pixels are located at the image border, Ω_{B} (i.e., pixels at all sides of the image rectangle; see Fig. 2). When Ω_{ROI} touches Ω_{B} or is outside the image, 0 is assigned; otherwise, 1 is assigned. In our implementation, we set the border width to 1 pixel.

$$c_B = \begin{cases} 0, & \text{if } |\Omega_{\mathrm{ROI}} \cap \Omega_{\mathrm{B}}| \geq 1 \text{ or } |\Omega_{\mathrm{ROI}}| = 0 \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

3.2 Secure source areas (rule 2)

Inpainting methods fill the ROI with pixels copied from the vicinity [8], [20], [26] or, otherwise, use them as a clue for improving the inpainting quality [22], [23]. For example, PatchMatch [20] finds pixel clusters at random locations near the ROI and copies such clusters to fill the ROI. Therefore, it is important to secure as many resource pixels, i.e., pixels in the vicinity of the ROI, as possible. A shortage of resource pixels leads to conspicuously repeated

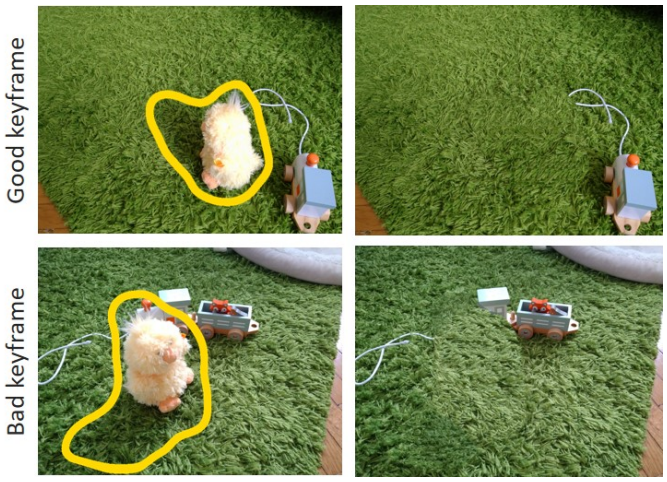


Fig. 5. "Secure similar pixel resources" (rule 3). The top and bottom rows show good and bad keyframes with respect to rule 3. While the green mat occupies the majority of the vicinity pixels in the top scene, the ROI touches the wooden locomotive in the bottom scene. Such a harmful configuration could be avoided by changing the viewpoint.

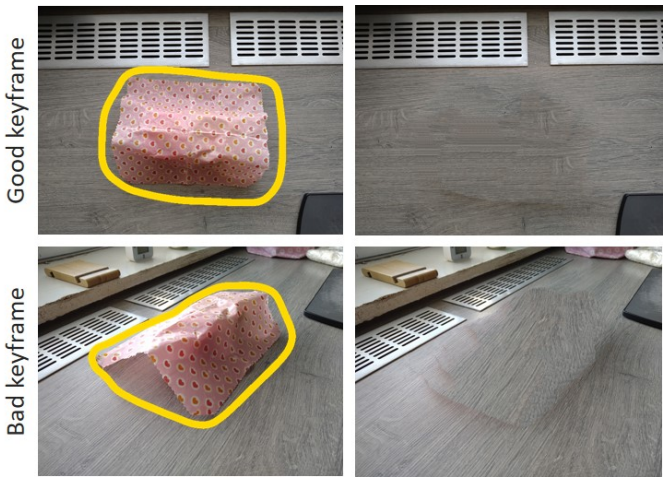


Fig. 6. "Minimize perspective distortion" (rule 4). The top and bottom rows show good and bad keyframes according to rule 4. The strong perspective distortion in the bad keyframe lets differently distorted pixel clusters jostle in the ROI.

patterns and other artifacts. Most DR inpainting methods perform poorly on views close to the diminished object, as the ROI occupies too much of the FOV. Fig. 4 shows two keyframe inpainting results with different ROI size in image space. The smaller ROI finds enough usable pixels in the source area (Fig. 4, top), while with the larger ROI almost occupies the entire FOV, forcing certain pixel chunks to appear repeatedly (Fig. 4, bottom).

To represent the balance of ROI and source region, we calculate a ratio $c_S \in [0,1]$ of the number of pixels of Ω_{ROI} over the number of pixels of the entire image, Ω_{FOV}:

$$c_{\mathrm{S}} = 1 - \frac{|\Omega_{\mathrm{ROI}}|}{|\Omega_{\mathrm{FOV}}|}. \tag{2}$$

Here, Ω_{FOV} is the entire FOV of the keyframe. This criterion indicates the relative source region size, $\Omega_S = \Omega_{\mathrm{FOV}} \setminus \Omega_{\mathrm{ROI}}$.

3.3 Secure similar pixel resources (rule 3)

Inpainting methods find similar texture patterns or similar neural features and reproduce them in a ROI, and such similar features should connect the boundaries of the ROI. Therefore, finding good matches depends on the availability of pixel resources which are similar to the pixels at the border of the ROI. Fig. 5 demonstrates how the choice of viewpoint influences what appears at the ROI border and, consequently, if rule 3 can be addressed. When the ROI is surrounded only by the pixels of the uniform green rug, the majority of the source area has similar pixels, making inpainting easy. Conversely, when the ROI touches distinct pixel clusters (e.g., the floor and the toy train), inpainting has difficulties connecting these clusters with other pixels.

To keep useful source pixels in the camera FOV, we measure similarities of pixels around the ROI borders to those in the other source pixels. We separate the source region into two regions: Ω_{S1} surrounds the ROI with a certain width. It is obtained by dilating the ROI 20 times with a 3×3 kernel and intersecting the dilation result with the source region. Ω_{S2} contains the remaining pixels.

$$\Omega_{\mathrm{S1}} = \Omega_{\mathrm{ROI}}^{\mathrm{Dilated}} \cap \Omega_{\mathrm{S}}, \qquad \Omega_{\mathrm{S2}} = \Omega_{\mathrm{S}} \setminus \Omega_{\mathrm{S1}} \tag{3}$$

We use the histogram correlation between pixels in Ω_{S1} and in Ω_{S2} to find pixels $\mathbf{p} \in \Omega_{\mathrm{S2}}$ that can potentially connect border pixels surrounding the ROI, i.e., $\mathbf{p} \in \Omega_{\mathrm{S2}}$. We calculate the pixel correlations in color $\mathcal{C}$ and normal $\mathcal{N}$ maps obtained by dense SLAM [25]. We denote such correlations as follows:

$$c_{\mathrm{C}} = (corr(H(\mathcal{C}_{S1}), H(\mathcal{C}_{S2})) + 1)/2, \tag{4}$$

$$c_{\mathrm{N}} = (corr(H(\mathcal{N}_{S1}), H(\mathcal{N}_{S2})) + 1)/2, \tag{5}$$

where $H(X)$ is a histogram of an image X, and $corr(H_1, H_2)$ calculates correlations between two image histograms H_1 and H_2. For normal vectors, we uniformly discretize the 3-channel values into 255 bins as in $\mathcal{C}$.

3.4 Minimize perspective distortion (rule 4)

It is challenging to handle perspective distortion in image inpainting without knowing scene geometry [27]. Some DR methods remove distortions via homography warping when a dominant plane is known from flat marker placement [2], [18] or from a sparse 3D reconstruction [9]. However, such an approach is limited to planar scenes. Without compensating for perspective, pixel clusters copied from a source region will appear in a different size and with arbitrary distortions. Also, depth variations in the FOV result in nonuniform pixel resolutions when the camera moves from the keyframe location, resulting in more noticeable interpolation artifacts [17]. Consequently, a keyframe should largely avoid all perspective distortions. Fig. 6 compares inpainting results with low (top) and high (bottom) distortion. Note how perspective distortion introduces unnaturally compressed textures at the back of the ROI.

When DR inpainting is applied to planar scenes [2], [9], the planes are warped to a canonical space facing towards the camera. In a similar manner, we prefer a keyframe

Fig. 7. "Avoid depth discontinuities" (rule 5) The top and bottom rows show good and bad keyframes according to rule 5. A large depth discontinuity wrongly connects the wall edges. The far wall on the left gnaws into the front wall on the right.

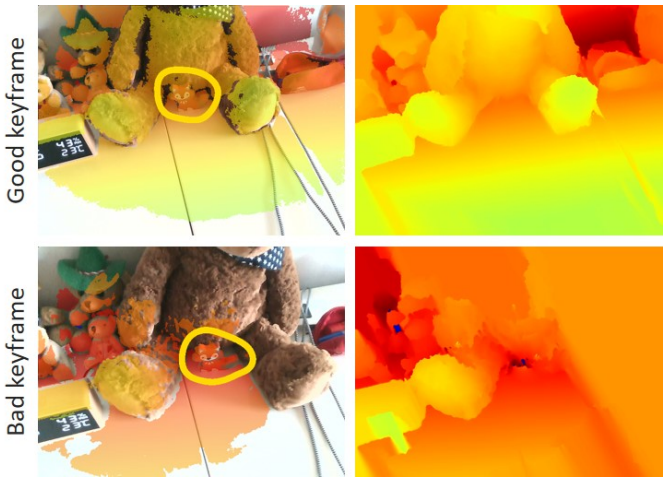


Fig. 8. "Reconstruct the scene fully" (rule 6). The top and bottom rows show good and bad keyframes with respect to rule 6. The left column shows keyframes with a SLAM reconstruction overlay. SLAM needs to explore the scene to collect sufficient depth samples. Note that pixels that have not been covered by SLAM are fully filled with a diffusion-based inpainting algorithm [32] before running the main depth inpainting [15]. However, real observations are more reliable than the convoluted depth taken from adjacent pixels.

containing many normal vectors facing the camera, thereby increasing the chances to observe pixels under low distortion. Therefore, we measure the fraction of scene normal vectors in Ω_S that are facing the camera. Our criterion

$$c_{\mathrm{Z}} = \frac{1}{|\Omega_S \cap \Omega_{\mathrm{R}}|} \sum_{\mathbf{p} \in \Omega_S \cap \Omega_{\mathrm{R}}} \mathcal{N}_z(\mathbf{p}) \tag{6}$$

sums the z-coordinate $\mathcal{N}_z(\mathbf{p})$ of a normal vector at a pixel $\mathbf{p}$, normalized to $[0, 1]$ by the number of eligible pixels.

Note that this metric is inspired by inpainting approaches for planar scenes [2], [9], [18]. In fact, we found that it performs as intended for 2D inpainting approaches [8], [20], [28] and counter-intuitively for a 3D inpainting approach that works in 3D scenes [15] (see Sections 3.7 and 5.3).

3.5 Avoid depth discontinuity (rule 5)

3D scene completion, especially from a single shot, is a challenging computer vision problem. Existing methods are limited either to simple textures and geometries or to small resolutions [29], [30]. Applying texture deformation after inpainting on a plane may handle smooth depth changes due to a curved surface or small bumps [19]. The state of the art in DR inpainting method applies RGBD inpainting in multiple keyframes by propagating the first inpainting results to the others [15]. However, more depth discontinuity requires more keyframe inpainting, leading to artifacts induced by incomplete pixel propagation. Therefore, depth continuities around ROI should be avoided. Fig. 7 compares inpainting results in keyframes with few depth differences around the ROI and with a strong depth discontinuity in the same scene.

We measure depth discontinuities in Ω_{S1} to find viewpoints where less depth discontinuities are observed. The depth discontinuity $d_{\mathcal{D}}$ is calculated according to the literature [31]. We count pixels having depth discontinuity and normalize the sum with the valid pixel count within Ω_{S1} to keep the value within 0 to 1, i.e., $c_D \in [0, 1]$,

$$c_{\mathrm{D}} = 1 - \frac{1}{|\Omega_{S1} \cap \Omega_{\mathrm{R}}|} \sum_{\mathbf{p} \in \Omega_{S1} \cap \Omega_{\mathrm{R}}} d_{\mathcal{D}}(\mathbf{p}), \tag{7}$$

$$\text{where } d_{\mathcal{D}}(\mathbf{p}) = \begin{cases} 1, & \text{if } \frac{|\nabla \mathcal{D}(\mathbf{p})|}{\mathcal{D}(\mathbf{p})} \geq t_D, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

and the threshold $t_D = 0.1$ was empirically determined.

3.6 Reconstruct the scene fully (rule 6)

We assume that scene reconstruction is obtained using SLAM. In this case, it is unrealistic to assume the 3D reconstruction will be complete. Thus, we must consider how much of the *relevant* scene has been covered by SLAM. Note that depth inpainting gets more reliable when diffusion-based depth inpainting [32] is applied before sampling the depth values from SLAM to finalize the depth inpainting [15]. However, diffusion-based depth inpainting becomes less accurate when the holes to fill in are large.

We label pixels in a keyframe candidate that are covered by any region of a projected scene reconstruction (i.e., SLAM map) in a frame as Ω_{R} (see Fig. 2 for an example) and use it for calculating the reconstruction reliability,

$$c_{\mathrm{R}} = \frac{|\Omega_{\mathrm{R}}|}{|\Omega_{\mathrm{FOV}}|}, \tag{9}$$

Fig. 8 compares two depth inpainting results with different reconstruction coverage. The sufficient reconstructed depths increase the chance to find smoother depth connections within the FOV (Fig. 8, top), while shortage of depth samples suffers from finding related depths (Fig. 8, bottom). Especially, when the surrounding pixels lack the depth values, those depth pixels are filled with the diffusion-based inpainting. As the Poisson solver-based depth inpainting [15], [33] connects depth gradients from the outer to the

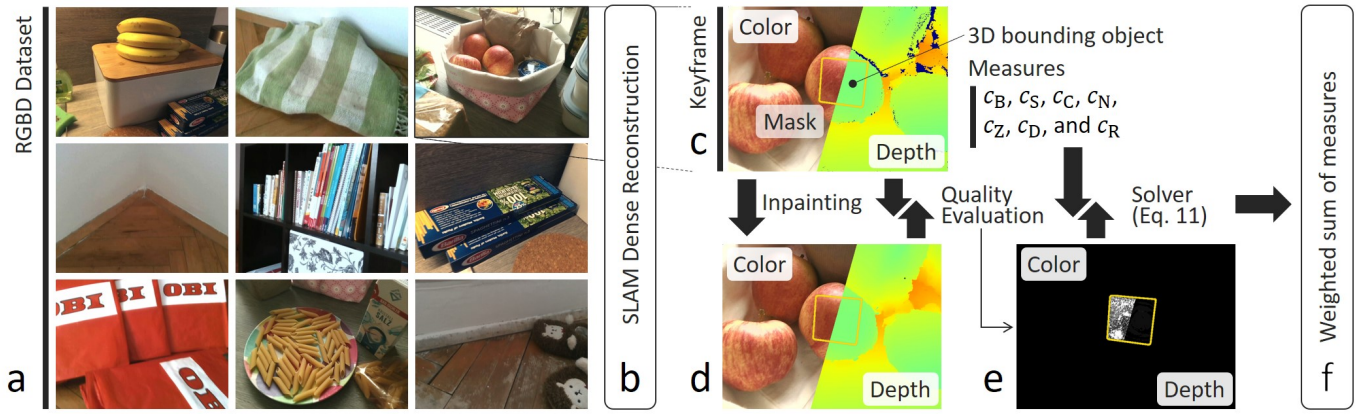


Fig. 9. Dataset and procedure for calculating the weights. (a) We collected pairs of color and depth images in nine scenes. As represented in the example frames, those scenes consist of various colors and geometries. (b) We run a SLAM method to obtain the global map in all scenes. (c) Each keyframe in the dataset is randomly selected and contains the calculated measures. A 3D bounding object is registered in the scene, and the projection of the object defines the ROI. (d) Given the color and depth images, the keyframe is inpainted. (e) The inpainted color and depth images are rated using a perceptual score (LPIPS), and the weights representing the contribution of each measure to the inpainting quality are calculated. (f) Finally, the resulting weights are used as balancing parameters of the measures.

inner ROI, surrounding depth pixels should have observed depth pixels. Therefore, checking valid depth pixels within Ω_{S2} would be an interesting future extension.

3.7 Balancing the Metrics

The six criteria need to be combined into a single metric c measuring the overall suitability of keyframes for DR. However, it is not clear how one criteria performs better than the others. Some may even work counter-intuitively because all change jointly from one frame to the next candidate. Therefore, we analyze the contributions of each criteria through an evaluation using a real dataset. Since the relationships between metrics are unknown, we use a simple and robust linear combination,

$$c = c_{\mathrm{B}} \cdot \sum_{k \in \{\mathrm{S,C,N,Z,D,R}\}} w_k c_k, \tag{10}$$

where w_k are the weighting parameters corresponding to our criteria, c_{S}, c_{C}, c_{N}, c_{Z}, c_{D}, and c_{R}. To determine the weighting parameters w_k, we collect pairs consisting of an RGBD image and a mask image. The RGBD images are considered the ground truth of inpainting results, as RGBD images with blank pixels in the masked area are the input to our inpainting system.

We choose LPIPS [34] as an image quality measure, as it mimics human perceptual behavior well. After running an inpainting method, we calculate v_{LPIPS} by comparing the ground truth image and the inpainted image in multiple scenes and conditions. We expect that those measures which strongly contribute to reducing the error will have high weights correlating with image quality. Therefore, we solve the following equation for the six unknown weights w_k that best describes $v_{\mathrm{score}} = 1 - v_{\mathrm{LPIPS}}$.

$$\left\| v_{\mathrm{score}} - \sum_{k \in \{\mathrm{S,C,N,Z,D,R}\}} w_k c_k \right\| \rightarrow \min \tag{11}$$

Fig. 9 illustrates the entire procedure. For depth image scores, we apply LPIPS to inverse depth images. The intention here is that we wish to evaluate the "plausibility" of inpainted depth values instead of absolute differences. This approach relies on how the depth images are visualized. We normalize each depth image by its minimum and maximum values and treat it as a grayscale image.

As discussed in literature [35], inpainted areas can show arbitrary content, as long as the content is acceptable. Therefore, there is no need for measuring absolute errors in this context. As no unique ground truth exists for a given inpainting problem, quality will be subjective and application-dependent. However, to obtain quantitative results, we use the original image content (before the mask is applied) as our reference image. Alternatively, one could employ human subjects to grade the quality of inpainting, using a Likert scale or a similar rating system. However, we take advantage of our automated approach, making applying our framework to different inpainting methods easier. We found that using the original image content as reference works well as long as the sample size is large enough.

4 Evaluations

We derive the weighting parameters using a real dataset and evaluate the effectiveness of the proposed metric to find good keyframes to inpaint. This section describes the weight computation, the collected dataset, and how we evaluate the proposed approach with the balanced weighting parameters by comparing it with baseline approaches.

4.1 System implementation

Our system consists of two modules, inpainting and SLAM. Inspired by the work of Mori et al. [15], we track and reconstruct scenes with a dense SLAM method [25] and inpaint a selected keyframe. The SLAM system provides depth and normal maps that we use as $\mathcal{D}$ and $\mathcal{N}$ described in the previous section. The inpainting algorithm finds an optimal transformation $f^* : \Omega_{\mathrm{ROI}} \rightarrow \Omega_S$ by minimizing the sum of squared difference (SSD) of color and normal patches, denoted as $\mathrm{R}_{\mathcal{C}}$ and $\mathrm{R}_{\mathcal{N}}$, respectively,

$$f^* = \arg\min_f \sum_{\mathbf{p} \in \Omega_{\mathrm{ROI}}} \alpha \mathrm{R}_{\mathcal{C}}(f, \mathbf{p}) + (1-\alpha)\mathrm{R}_{\mathcal{N}}(f, \mathbf{p}), \tag{12}$$

TABLE 1
Statistics of the measures of 360 keyframes in nine scenes.

Measures	Statistics			
	Mean	Std. dev.	Min	Max
Source ratio	0.94	0.07	0.56	1.00
Color correlation	0.72	0.13	0.45	0.97
Normal correlation	0.75	0.12	0.56	0.99
Facing normal ratio	0.63	0.09	0.33	0.86
Depth continuity	0.97	0.04	0.84	1.00
Reconstruction reliability	0.83	0.11	0.40	1.00
Color variation [36]	50.4	20.3	19.4	106.0
Gaussian curvature [37] ($\times 10^{-5}$)	-0.96	0.97	-8.1	0.14

Both color and normal channels are normalized to the $[0, 1]$ range and balanced with a fixed weight $\alpha = 0.95$. We recover the inpainted depth map described by f^* from the depth gradient samples [15]. Compared to PixMix [8], we excluded the spatial term and added a random sampling step [20] to make the pixel search more flexible.

We tested our system on a notebook computer (Intel Core i7-6567U with 3.3 GHz, 16 GB RAM, external NVIDIA GeForce GTX1080Ti connected via Thunderbolt 3, Windows 10). The calculation of the measures takes 10.3 ms on average for an image at 640×480 pixel resolution (c_{B}: 0.14 ms, c_{S}: 0.01 ms, c_{C}: 0.89 ms, c_{B}: 4.35 ms, c_{B}: 2.98 ms, c_{B}: 1.85 ms, c_{B}: 0.08 ms).

4.2 Dataset

We used Intel RealSense SR300 as an RGBD sensor to record our dataset. After the recording, we smoothed the depth and normal maps over all frames in each scene to obtain $\mathcal{D}$ and $\mathcal{N}$. For each image, we generated a mask of a registered 3D bounding box overlaid in the scene. We manually placed the 3D bounding box in each scene to ensure that reasonable pixel sources were available and could be found by our algorithm (i.e., we did not include any pathological examples). We prepared two variations of the 3D bounding box by changing size, position, and orientation in each scene.

Then, we randomly selected 20 keyframes per scene and per bounding box condition and generated 100 inpainting results per keyframe to minimize variations in inpainting results due to random seed placement. Consequently, we obtained $36,000$ keyframe inpainting results in total (9 scenes $\times$ 2 mask object conditions $\times$ 20 keyframes $\times$ 100 inpainting results). TABLE 1 summarizes our dataset.

4.3 Analysis of weighting

Fig. 10 summarizes the calculated weights across all combinations of measured parameters. Using a Pearson correlation analysis we found that color and depth scores are highly correlated ($\rho = 0.67$, $p < 0.01$). Therefore, we average the scores for color and depth to determine the final weight.

As often discussed in the literature on inpainting, the source ratio (rule 2) has the second highest ($w_{\mathrm{S}} = 0.44$) and the highest impact ($w_{\mathrm{S}} = 0.58$) on the inpainting quality in color and depth domain. In other words, the inpainting method is able to offer high-quality keyframe inpainting when a small ROI is given. Except for this rule 2 and

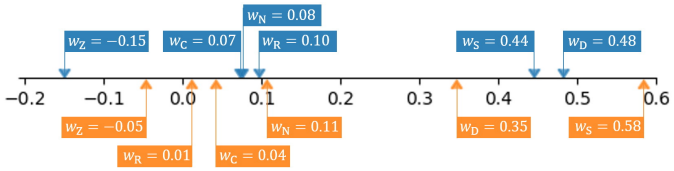


Fig. 10. Weights over all measures. We use the weighting parameters to balance the metric. We obtained the weights for color (blue) and depth (orange).

depth continuity (rule 5, $w_{\mathrm{D}} = 0.48$ and 0.35 in color and depth, respectively), the other measures show relatively low weights in comparison. This suggests that there is no clear guideline to find a "good" keyframe that provides high inpainting quality, apart from the source ratio and depth continuity. This result explains the difficulties encountered in manually selecting keyframes and supports the idea of an automatic approach. While meeting individual criteria may have only a subtle impact, addressing many or all of them can increase the odds of selecting a good keyframe.

The weight of rule 4 (the ratio of normals facing the camera) appears counter-intuitive, as it takes negative values in both color and depth domains ($w_{\mathrm{Z}} = -0.15$ and -0.05). As discussed in Section 3.4, the rule applies best to inpainting algorithms for planar scenes. However, our test dataset contains a variety of depth discontinuities, which causes varying surface normals (see the "Gaussian curvature" score in TABLE 1 and example images in Fig. 9a). As demonstrated in Fig. 18, w_{Z} shows the expected positive values for planar inpainting algorithms.

Overall, our results suggest that, in keyframe selection, one has to keep the ROI size small and also maintain color, depth, and normal values of the observed scene. These requirements are difficult for users to satisfy concurrently, especially when moving the camera, while the scene reconstruction is ongoing. When the user is not able to move in order to change the ROI size, the keyframe selection becomes hard to judge. In any case, updates in 3D reconstruction are often hidden from the user in an augmented reality (AR) view, and, therefore, are hard to be interpreted as keyframe selection criteria. Thus, we implemented applications including automatic keyframe selection (Section 5.1) and user navigation for an appropriate keyframe selection (Section 5.2).

4.4 Comparisons with a baseline approach

To evaluate the weighted metric, c, we prepared another dataset by including seven new sequences using different scenes. Like the previous dataset (Section 4.2), the new dataset consists of color, mask, depth, and normal images. Each scene contains 550 frames. We excluded the first 50 frames as extreme cases since they are erratic due to the SLAM initialization. We further separated each scene of 500 frames into five subset sequences containing the first 100, 200, 300, 400, and 500 to evaluate shorter and longer sequences. Using the new dataset, we compared our approach, named *weighted selection* (WS), with a baseline, named *average selection* (AS), which sets all weights uniformly to $w_k = 1/6$. Both contenders selected their best and worst keyframes after scoring all frames. We then ran our

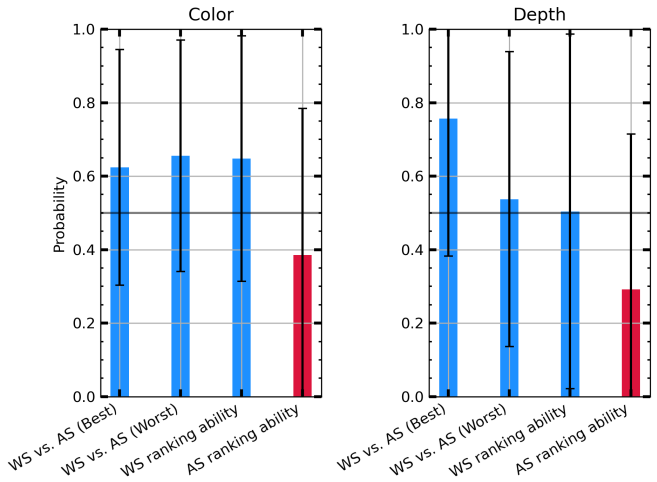


Fig. 11. Selection quality of WS and AS in color and depth domains. WS is able to select keyframes with higher chances for good inpainting results in terms of LPIPS. In comparison, AS has a more limited ability with respect to selection of ranking.

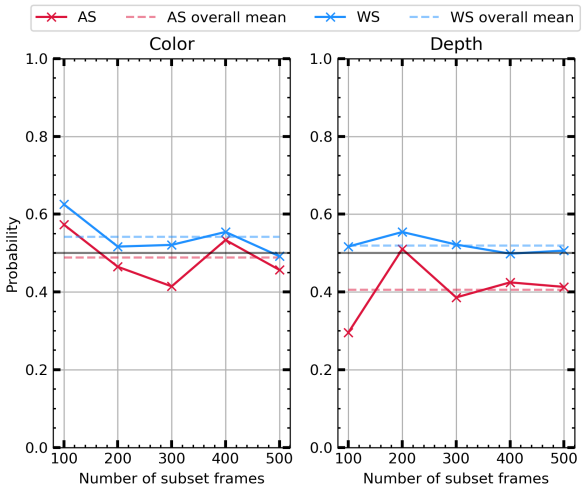


Fig. 12. Comparison of RS, WS and AS in the color and depth domains. The sampled points show the probability that LPIPS scores of the best-selected keyframes of WS and AS are lower than the random selection (RS) in each sub-sequence.

keyframe inpainting on the selected keyframes 100 times and calculated the LPIPS scores to evaluate how stable the inpainting method's quality is.

While we expect that AS selects plausible keyframes that follow the rules in Section 3, WS should improve chances to obtain higher-quality inpainting results, i.e., lower LPIPS scores, on the selected best keyframes. Therefore, we calculate the probability that LPIPS scores of WS are lower than those of AS in their best-scored keyframes per sub-sequence. In addition, we expect that WS selects keyframes that result in lower-quality inpainting results on the selected worst-scored keyframes. We therefore calculate the probability that LPIPS scores of WS are higher than those of AS in their worst-scored keyframes. We also evaluate the ranking ability of AS and WS by checking if the approaches correctly evaluated the best keyframes better than the worst keyframes in LPIPS. For this purpose, we calculate the probability that LPIPS scores of AS and WS on their best keyframes are lower than those of their worst keyframes.

Fig. 11 shows the results in the seven test sequences with five subsets each. This figure summarizes that WS can select better and worse keyframes that would end up with higher and lower quality inpainting results, respectively, under its control (See the blue bars that go above the 50% borderlines in the figure). The ranking ability of AS is clearly worse than that of WS, which means that AS tends to confuse good and bad keyframes. The ranking ability of WS in the depth domain is relatively lower than that in the color domain. Balancing the quality between the color and depth inpainting remains future work.

4.5 Comparisons with random selections

One may consider that, as novice users would do, randomly picking keyframes might work better than our selections. We, therefore, randomly selected 100 keyframes in each of the seven test sequences and inpainted the keyframes 100 times each. We binned these keyframes into each subset. Then, we compared the probability that LPIPS scores of best keyframes are lower than those of the random selection in each sub-sequence. We labeled this approach *random selection*, or RS.

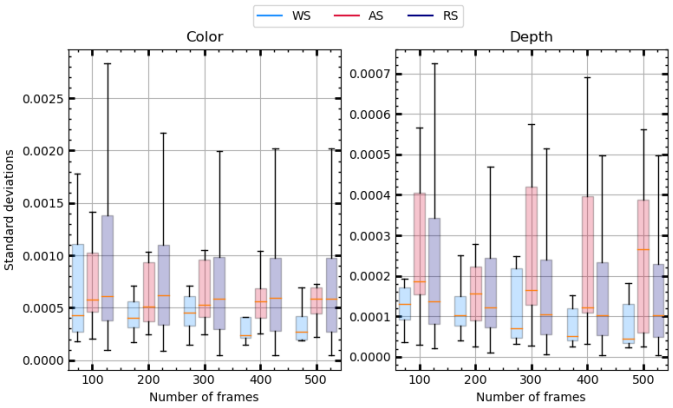
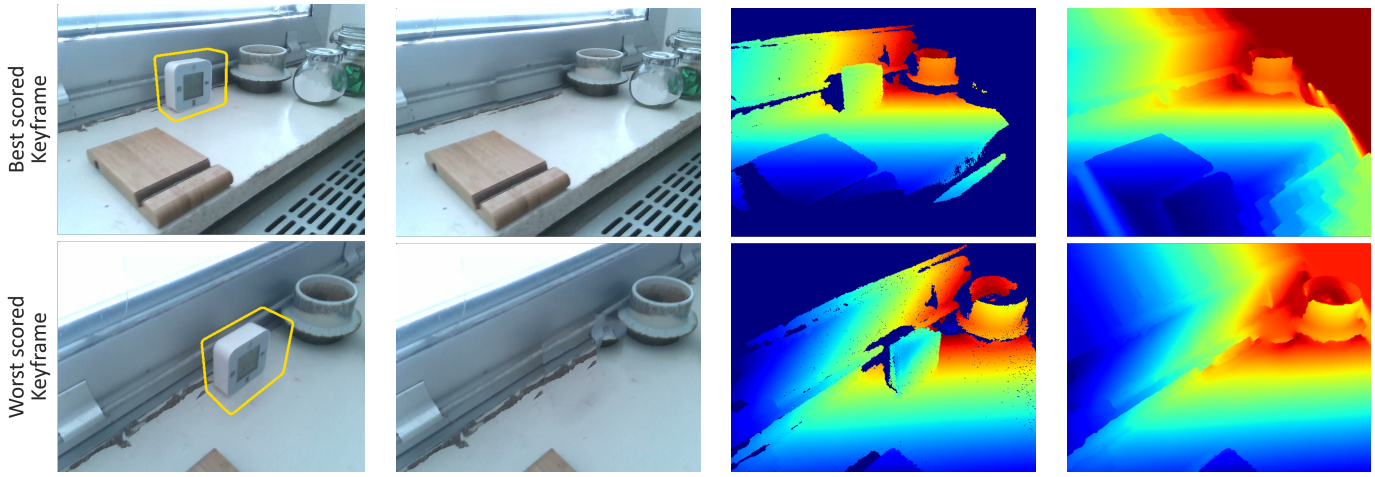


Fig. 13. Standard deviations of WS, AS, and RS in each sub-sequence. The lower values mean smaller differences in inpainting results.

Fig. 12 summarizes the mean probability of how often the best keyframes of AS and WS perform better than RS in each sub-sequence of the seven scenes. WS clearly achieves higher performance than AS in all sub-sequences, especially in the depth channel. WS achieves almost identical performance to RS in longer sequences. However, we note that RS has no ability of ranking, while WS does (Fig. 11). Overall, WS performs better (54.2% and 51.9%) than AS (48.8% and 40.6%) and RS (50% and 50%) in the color and depth domain, respectively.

Fig. 13 shows variations in standard deviations of LPIPS scores in each sub-sequence. Here, standard deviations represent possible variances in inpainting results at each selected keyframe. Therefore, a lower score implies smaller differences in inpainting quality – the user may expect fewer trials until successful inpainting. We see that WS achieves the best performance in this evaluation.

(a) Frames progressing from the left to the right



(b) Original color + Mask area (c) Inpainted color (d) Original depth (e) Inpainted depth

Fig. 14. Automatic keyframe selection. (a) To an untrained user, it may not always be immediately obvious which frame ends up with a high-quality inpainting result. The proposed system evaluates the criteria over the frames to automatically select a good keyframe. The user explores the scene taken under various camera motions, e.g., while reconstructing the scene for AR. (b–e) Our keyframe criteria avoids selecting frames that would result in low quality results and increases chances to obtain a high-quality inpainting both in color and depth domains.

5 Applications

We demonstrate how our metric can be used to ease keyframe selections in practical AR applications. The first application is a complete DR system with an automatic keyframe selection function. The second application is an AR navigation to let the user seek better keyframes. The third application is an AR marker hiding tool, which shows how our method can be applied for planar scenes with a known fiducial marker.

5.1 Automatic keyframe selection

Upon the object of interest identification, a DR system can immediately start calculating the metric. While the user explores the scene, the system updates the highest frame score and keeps the frame as a keyframe (color, depth, and normal images, a pose $\mathbf{T} \in \mathbb{SE}(3)$, and a metric score c). Our automatic keyframe selection scheme preserves such a keyframe and inpaints it on the user's request, independent of the current camera location.

Fig. 14 presents an example case of our automatic keyframe selection. While the camera explores the scene, the proposed approach calculates all criteria in Section 3 and the resulting metric, c, at every frame (Fig. 15). As the camera perspective varies during the exploration, it is difficult for the user to determine which specific frame to choose as a keyframe by visual checks alone (Fig. 14a). The proposed approach is able to automatically find a keyframe where the inpainting algorithm can be expected to provide overall good results in both color and depth domains, while avoiding poor keyframe selections (Fig. 14b–e).

Sections 4.4 and 4.5 suggest that our method may, in rare cases, select a poor keyframe, where inpainting repeatedly

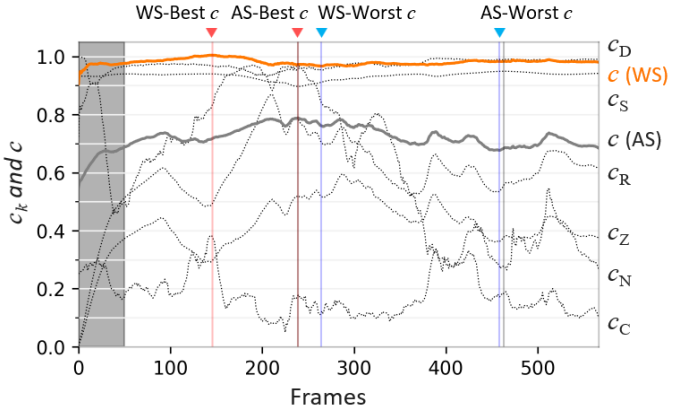


Fig. 15. Plots of metrics over the frames of the scene in Fig. 14. The DR system evaluates every frame, once a bounding object is placed in the scene. During the scene exploration, the system preserves best-scoring keyframes, while discarding others. Therefore, regardless where the user is located, the DR system can start keyframe inpainting with the latest keyframe. The black bars, which almost overlap on the AS selections, show the best and worst keyframe selected solely by c_S.

fails. To avoid such worst-case behavior, we rely on the ranking and switch to the second best keyframe after several failed inpainting attempts. To this end, we keep multiple keyframes, e.g., on the second peak after around #400 frame in Fig. 15. First, we compare the current frame score c with the score c_i of keyframe i. If c is greater than c_i, and a Frobenius norm $||\mathbf{I} - \mathbf{T}_i^{-1}\mathbf{T}||_F$ is smaller than a threshold t[1], then keyframe i is replaced with the current frame. If the Frobenius norm is greater than or equal to t, we check

1. The threshold t scales depending on the SLAM trajectory scale.

(a) Scene exploration (b) 3D arrow overlay on the AR display

Fig. 16. User navigation. (a) Our method evaluates frames only along the camera trajectory. To explore better keyframes, the AR device should be moved to $\mathbf{t}_i$, where the highest score was obtained. (b) To this end, the AR navigation guides the user with a 3D arrow annotation.

keyframe $i+1$ and repeat this procedure until the last keyframe.

One may consider that our final metric, c, relies too much on c_{S} (rule 2). The parameter c_{S} does have a more substantial effect than the others, but its contribution is only proportional to the relative chosen weight as discussed in Section 4.3. See Fig. 15 for a visual analysis.

5.2 User navigation

To support easier explorations in the automatic keyframe selection system, the DR system encourages the user to move the camera randomly, while the system accumulates the scores over the frames. Then, the system can guide the user to the location with the highest score to identify an even better keyframe. To this end, we implemented an AR navigation visualization that shows a 3D arrow to indicate the direction to the best keyframe at $\mathbf{t}_i \in \mathbb{R}^3$ from the current frame at $\mathbf{t} \in \mathbb{R}^3$. The 3D arrow faces towards $(\mathbf{t}_i - \mathbf{t})/||\mathbf{t}_i - \mathbf{t}||$. Fig. 16 illustrates the user navigation application.

5.3 Marker hiding

Marker hiding is a popular application of DR, which inpaints a fiducial marker used for tracking the camera [2], [18]. We implement a marker hiding application and use it to compare multiple 2D inpainting algorithms for our approach. We expect different weighting parameters, as every inpainting method has a distinct statistical behavior.

Substituting different inpainting methods. In this application, we replace the SLAM system [25] with an ArUco marker [38] to detect camera poses at every frame. For comparison, we selected PixMix [8], a successful algorithm for inpainting, and the content-aware fill in Adobe Photoshop CS5.1, representing the commercial state of the art. We assume that the marker is placed in a planar scene. Therefore, we make the following assumptions for the criteria in Section 3: Normal correlation is always the same, i.e., $w_{\mathrm{N}} = 0$ (rule 3). The scene normal vector is equivalent to the up vector of the marker (rule 4). No depth continuity

(a) Marker image (b) Ground truth image

Fig. 17. Example frames from the AR marker hiding dataset. The dataset consists of (a) marker images and (b) their corresponding ground truth images. Marker up vectors are used to calculate eq. 6. We inpaint the ground truth image with a given ROI.

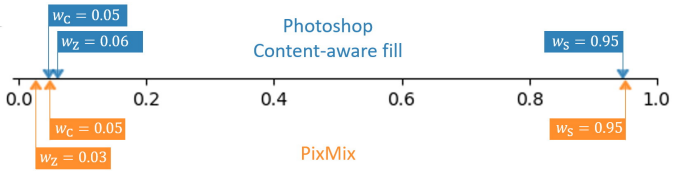


Fig. 18. Weights for the AR marker hiding application using (top) PixMix inpainting and (bottom) Photoshop content-aware fill as the core inpainting algorithm.

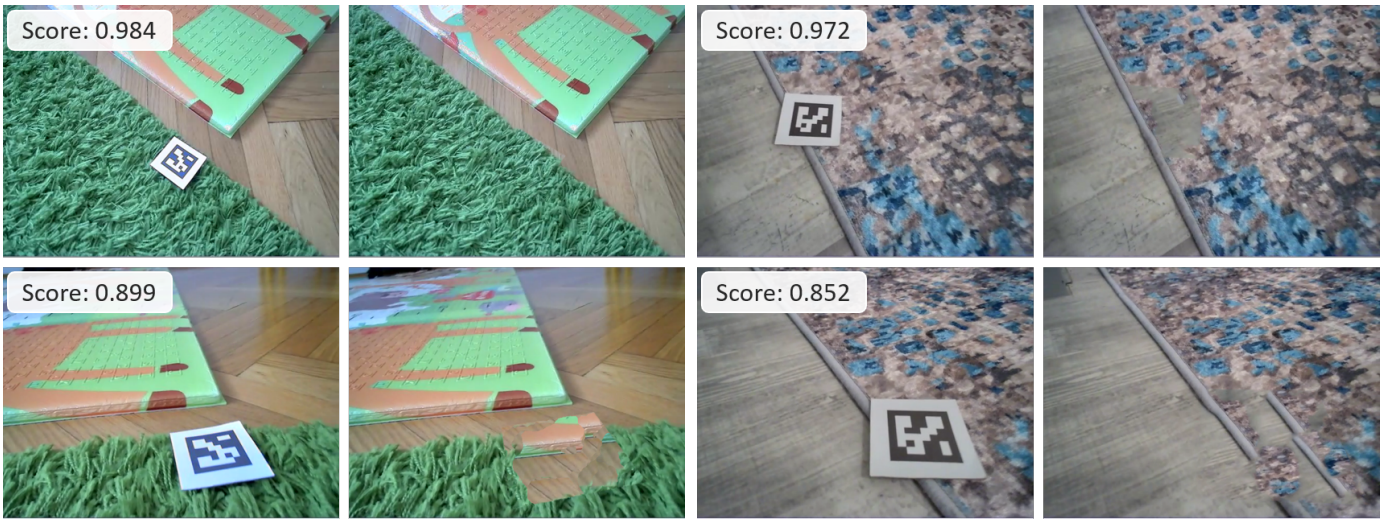
is observed, i.e., $w_{\mathrm{D}} = 0$ (rule 5). The scene geometry is complete and corresponds to the marker plane, i.e., $w_{\mathrm{R}} = 0$ (rule 6).

Dataset generation. We prepared another dataset that includes 127 pairs of images with and without markers by fixing a camera while photographing (Fig. 17a, b). On each keyframe, we ran the inpainting algorithms 100 times to obtain 127,000 inpainted results in total for each algorithm. The ROI is identified by a marker as a slightly larger rectangle area than the actual marker to ensure the marker area is fully covered (Fig. 17a).

Results. Fig. 18 shows weights of the two inpainting algorithms. Both algorithms show strong impact to the source size (rule 2), i.e., c_{S}. The resulting weights are similar to each other, but different from the 3D inpainting case (especially, $w_{\mathrm{Z}} \leq 0$). The order of w_{Z} and w_{C} is different in the inpainting algorithms, but the differences are subtle. The obtained weights can be used to calculate the scores in AR marker hiding. Fig. 19 shows a qualitative comparison of a good and a bad keyframe inpainting in a scene based on our metric and weights. We show PixMix and Photoshop content-aware fill in Fig. 19a and b, respectively. Note that, as discussed in Section 3.4, with a known fiducial marker, we could transform the input image to a canonical image space to remove perspective distortions [2], [18], which diminishes the contribution of c_{Z}. However, for more generalized target objects [8], the inpainting algorithms would perform better on keyframes found with c_{Z} measured.

6 Discussion and Conclusions

In this paper, we studied a metric for *good keyframes to inpaint* in DR. The metric is calculated as a sum of six criteria, added according to the weights observed from inpainting real scenes. The results demonstrate that our



(a) PixMix inpainting (b) Photoshop content-aware fill

Fig. 19. Example frames of marker hiding inpainting results using (a) PixMix inpainting [8] and (b) Photoshop content-aware fill with a good and a bad keyframe, as distinguished using our metric, c, in a scene. Left and right column images are original and inpainted images, respectively.

method enables the automatic selection of keyframes for high-quality inpainting, reducing the number of inpainting attempts required until a convincing result is obtained. We further implemented three practical DR applications that demonstrate the impact of the proposed method.

While the proposed method improves DR experiences, some issues need to be addressed. We evaluate only keyframes, assuming that a well-inpainted keyframe would be used in subsequent frames for temporally coherent inpainting [8], [9], [15]. Due to the large contribution of c_S to the final keyframe score, a keyframe could suffer from its limited pixel resolution when used for the subsequent frames. However, existing inpainting algorithms for DR already mitigate this issue by successive pixel searching [8], [33] and multi-keyframe projection [15]. Nonetheless, one could extend our framework to evaluate not only individual keyframes but also multiple keyframes or entire frame sequences. Intervals between keyframes and computational load would also play an important role in the best set of keyframes selection. For example, too closely spaced keyframes merely waste performance while producing similar inpainting results.

To determine the balancing weights, we used LPIPS in our evaluations, which should closely mimic human perception. As mentioned in Section 3.7, this evaluation could be replaced with data collection from a user study for more human-centric results. However, evaluating inpainted images consistently in user studies is known to be difficult, as results depend on human judgment with respect to the application requirements [35]. We consider finding better ways of normalizing such judgment an interesting, but challenging direction for future work. Alternatively, developing a deep neural network that scores frames directly could circumvent the problem. However, the weight analysis would still remain a black box.

Acknowledgments

This work was enabled by the Austrian Science Fund FWF (grant no. P33634).

References

[1] D. Schmalstieg and T. Höllerer, *Augmented Reality: Principles and Practice*. Addison-Wesley Professional, 2016.
[2] O. Korkalo, M. Aittala, and S. Siltanen, "Light-weight marker hiding for augmented reality," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2010, pp. 247–248.
[3] F. Cosco, C. Garre, F. Bruno, M. Muzzupappa, and M. A. Otaduy, "Visuo-haptic mixed reality with unobstructed tool-hand integration," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 19, no. 1, pp. 159–172, 2013.
[4] D. Kalkofen, M. Tatzgern, and D. Schmalstieg, "Explosion diagrams in augmented reality," in *Proc. IEEE Virtual Reality (VR)*, 2009, pp. 71–78.
[5] S. Zokai, J. Esteve, Y. Genc, and N. Navab, "Multiview paraperspective projection model for diminished reality," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2003, pp. 217–226.
[6] P. Barnum, Y. Sheikh, A. Datta, and T. Kanade, "Dynamic seethroughs: Synthesizing hidden views of moving objects," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009, pp. 111—114.
[7] S. Siltanen, "Texture generation over the marker area," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2006, pp. 253–254.
[8] J. Herling and W. Broll, "High-quality real-time video inpainting with pixmix," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 20, no. 6, pp. 866–879, 2014.
[9] N. Kawai, T. Sato, and N. Yokoya, "Diminished reality based on image inpainting considering background geometry," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 22, no. 3, pp. 1236–1247, 2016.
[10] J. Shi and C. Tomasi, "Good features to track," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.
[11] S. Mori, S. Ikeda, and H. Saito, "A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 17, 2017.
[12] F. Rameau, H. Ha, K. Joo, J. Choi, K. Park, and I. S. Kweon, "A real-time augmented reality system to see-through cars," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 22, no. 11, pp. 2395–2404, 2016.

[13] S. Jarusirisawad, T. Hosokawa, and H. Saito, "Diminished reality using plane-sweep algorithm with weakly-calibrated cameras," *Progress in Informatics*, vol. 7, pp. 11–20, 2010.
[14] Z. Li, Y. Wang, J. Guo, L.-F. Cheong, and S. Z. Zhou, "Diminished reality using appearance and 3D geometry of internet photo collections," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 11—19.
[15] S. Mori, O. Erat, W. Broll, H. Saito, D. Schmalstieg, and D. Kalkofen, "InpaintFusion: Incremental RGB-D inpainting for 3D scenes," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 26, no. 10, pp. 2994–3007, 2020.
[16] S. Siltanen, "Diminished reality for augmented reality interior design," *The Visual Computer*, vol. 33, pp. 193–208, 2015.
[17] J. Philip and G. Drettakis, "Plane-based multi-view inpainting for image-based rendering in large scenes," in *Proc. ACM Symposium on Interactive 3D Graphics and Games (I3D)*, 2018.
[18] N. Kawai, M. Yamasaki, T. Sato, and N. Yokoya, "Diminished reality for ar marker hiding based on image inpainting with reflection of luminance changes," *ITE Transactions on Media Technology and Applications*, vol. 1, no. 4, pp. 343–353, 2013.
[19] N. Kawai, T. Sato, Y. Nakashima, and N. Yokoya, "Augmented reality marker hiding with texture deformation," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 23, no. 10, pp. 2288–2300, 2017.
[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, 2009.
[21] O. Whyte, J. Sivic, and A. Zisserman, "Get out of my picture! Internet-based inpainting," in *Proc. British Machine Vision Conference (BMVC)*, vol. 2, no. 4, 2009, p. 5.
[22] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5505–5514.
[23] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *International Journal of Computer Vision (IJCV)*, vol. 128, pp. 1867–1888, 2020.
[24] J. Herling and W. Broll, "Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2010, pp. 207–212.
[25] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *Proc. International Conference on 3D Vision*, 2013, pp. 1–8.
[26] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing (TIP)*, vol. 13, no. 9, pp. 1200–1212, 2004.
[27] H. Sasao, N. Kawai, T. Sato, and N. Yokoya, "A study on effect of automatic perspective correction on exemplar-based image inpainting," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 1, pp. 21–32, 2016.
[28] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *Proc. European Conference on Computer Vision (ECCV)*, 2010, pp. 29–43.
[29] R. Fujii, R. Hachiuma, and H. Saito, "Rgb-d image inpainting using generative adversarial network with a late fusion approach," in *Proc. International Conference on Augmented Reality, Virtual Reality and Computer Graphics (SALENTO AVR)*. Springer, 2020, pp. 440–451.
[30] H. Dhamo, K. Tateno, I. Laina, N. Navab, and F. Tombari, "Peeking behind objects: Layered depth prediction from a single image," *Pattern Recognition Letters*, vol. 125, pp. 333–340, 2019.
[31] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. European Conference on Computer Vision (ECCV)*, 2012, pp. 611–625.
[32] T. Schöps, M. R. Oswald, P. Speciale, S. Yang, and M. Pollefeys, "Real-time view correction for mobile devices," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 23, no. 11, pp. 2455–2462, 2017.
[33] S. Mori, J. Herling, W. Broll, N. Kawai, H. Saito, D. Schmalstieg, and D. Kalkofen, "3D PixMix: Image inpainting in 3D environments," in *Proc. International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2018, pp. 1–2.
[34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[35] D. Pathak, P. Krähenbühl, J. Donahue, and T. Darrell, "Context encoders: Feature learning by inpainting," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544.
[36] D. Hasler and S. E. Suesstrunk, "Measuring colorfulness in natural images," in *Human Vision and Electronic Imaging VIII*, vol. 5007, 2003, pp. 87–95.
[37] C. Zhao, D. Zhao, and Y. Chen, "Simplified gaussian and mean curvatures to range image segmentation," in *Proc. International Conference on Pattern Recognition (ICPR)*, 1996, pp. 427–431.
[38] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

Shohei Mori is a postdoctoral researcher at Graz University of Technology. His focus lies in augmented and diminished reality and the related computer vision and display technologies. He received B.S. (2011), M.S. (2013), and PhD (2016) in engineering at Ritsumeikan University, Japan. He received two Research Fellowships for Young Scientists (DC-1 & PD) from the Japan Society for the Promotion of Science.

Dieter Schmalstieg is a Professor at Graz University of Technology, Austria. His research interests are augmented reality, virtual reality, computer graphics, visualization and human-computer interaction. He received a PhD from Vienna University of Technology. He is recipient of the IEEE Virtual Reality technical achievement award, the IEEE ISMAR Career Award, and a Fellow of the IEEE.

Denis Kalkofen is an Assistant Professor at the Institute of Computer Graphics and Vision (ICG) at Graz University of Technology, Austria. Before joining ICG, he was a member of the Virtual Reality Laboratory at University of Michigan. His research is focused on visual computing for developing visualization, interaction, display and authoring techniques for Mixed Reality.