INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG04
MPEG VIDEO CODING

ISO/IEC JTC 1/SC 29/WG 04 m59517
April 2022, Online

Title: On increasing the subjective quality of posetraces for low bitrates

Source: Adrian Dziembowski, Dawid Mieloch, Poznań University of Technology

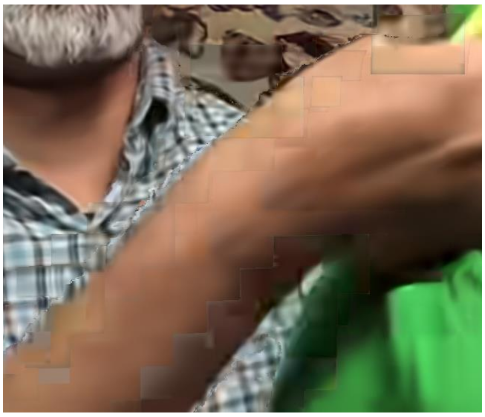
Abstract

This document presents a description of three changes increasing the subjective quality of posetraces for lower bitrates. Proposed modifications include changing the MinPatchSize parameter and two code changes in the renderer: skipping boundaries of patches and morphological filtration of the viewport's depth map.

1 Proposed modifications

1.1 Increasing the MinPatchSize

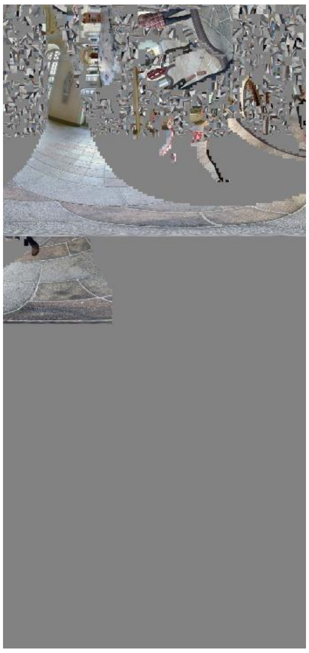
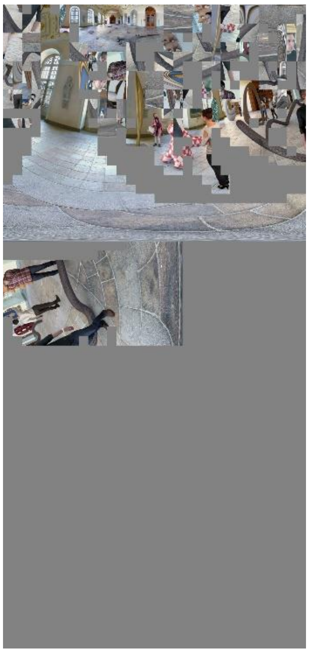
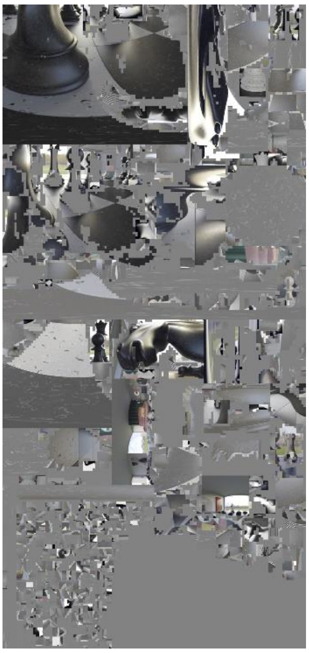
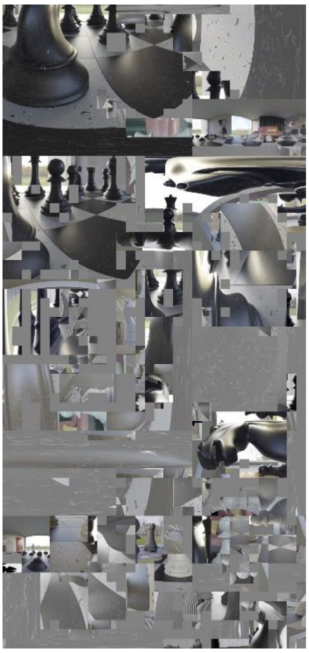
Typically, for low bitrates, various patches may slightly change their color and become visible being particularly annoying for a viewer (Fig. below).



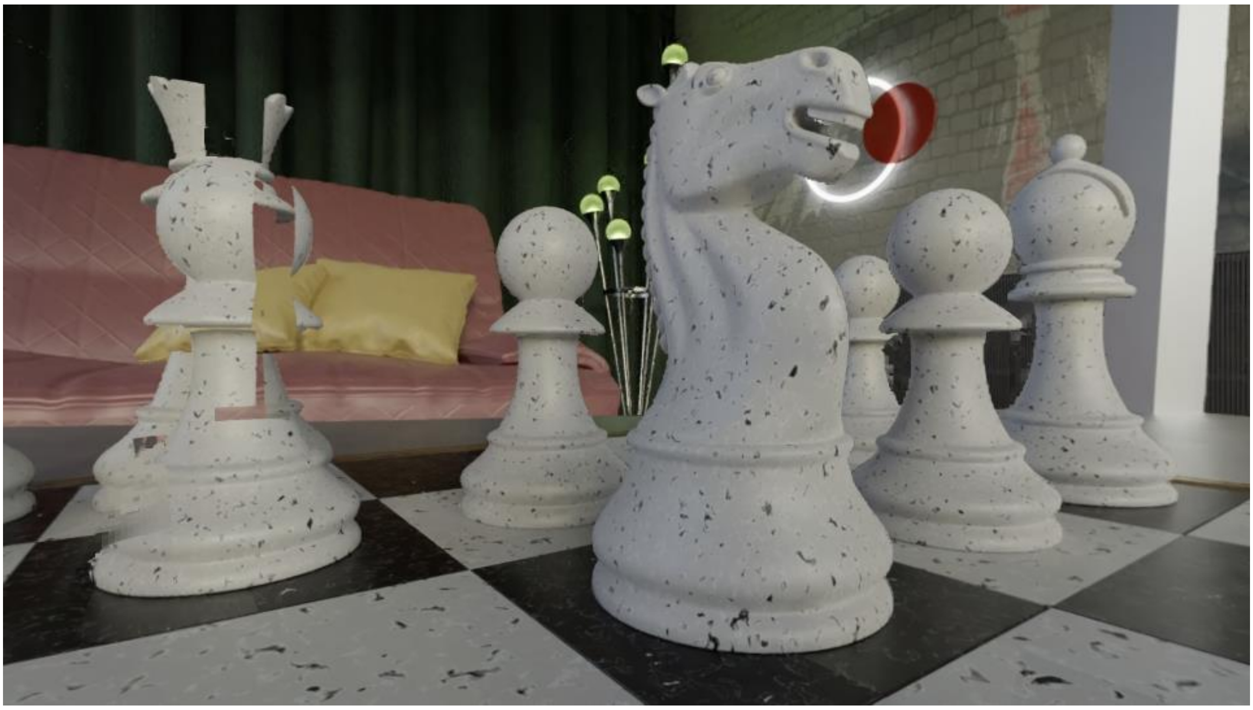
We propose to change the default patch size from [16, 32] to [64, 128]. Such a change has two major advantages:

- there are fewer edges between patches and the non-occupied area,
- occupied area is more temporally stable.

Both advantages allow to decrease bitrate or increase quality while preserving the same bitrate.

SB, t01, f52, Default MinPatchSize	SB, t01, f52, Modified MinPatchSize	SN, t01, f52, Default MinPatchSize	SN, t01, f52, Modified MinPatchSize
			

The disadvantage of the proposed change is visible in Fig. above, where non-pruned information for SN does not fit into the atlas anymore, resulting in some holes in the synthesized views:



1.2 Patch margin skipping

Even if there are fewer patches and their area is larger, the problem with visible patches persists, as their edges are still being destroyed by the video encoder at higher QPs.

To reduce this problem, we propose to skip the boundary of each patch while rendering. In this case, atlases look exactly the same and the bitrates are not affected. However, the edges of patches are not reprojected at the decoder side, decreasing the visibility of patch boundaries. We believe, that skipping four rows from top and bottom, and four most-left and most-right columns of each patch is an optimal solution.

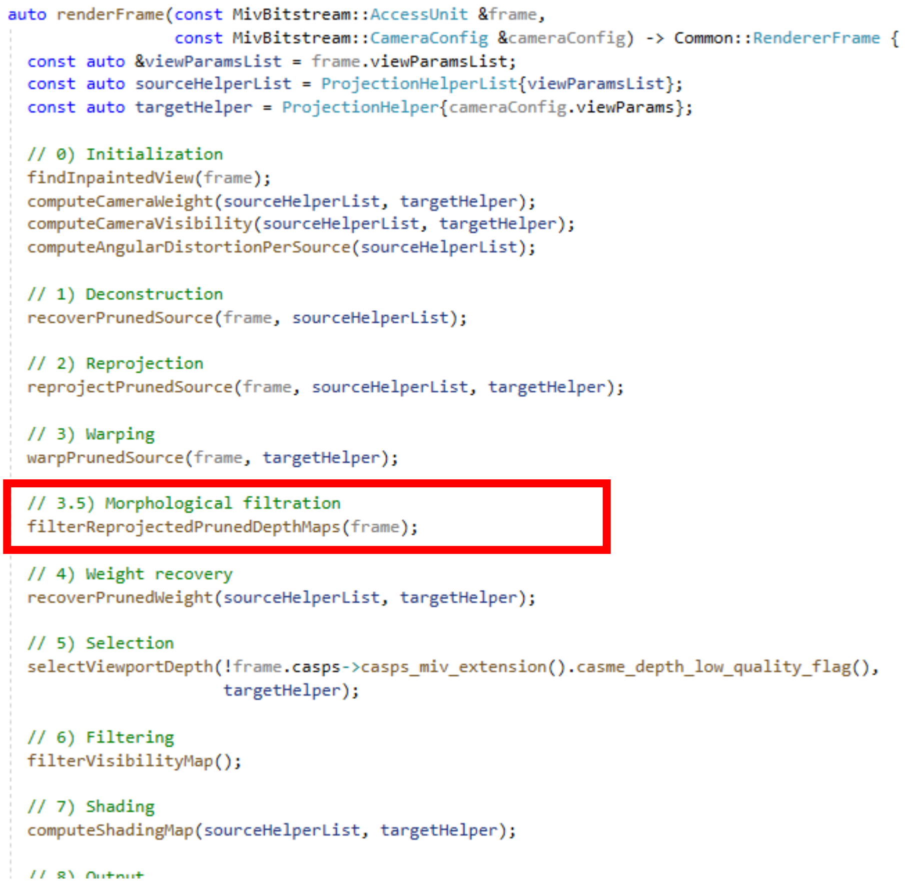
1.3 Morphological filtration of the viewport’s depth map

The third proposed modification is to enable morphological filtering of the reprojected depth map of the virtual view.

In proposed filtering, the viewport’s depth map is filtered by the morphological erosion filter with a 3x3 mask, performed 5 times. Then, 5 iterations of morphological dilation with the same mask are performed.

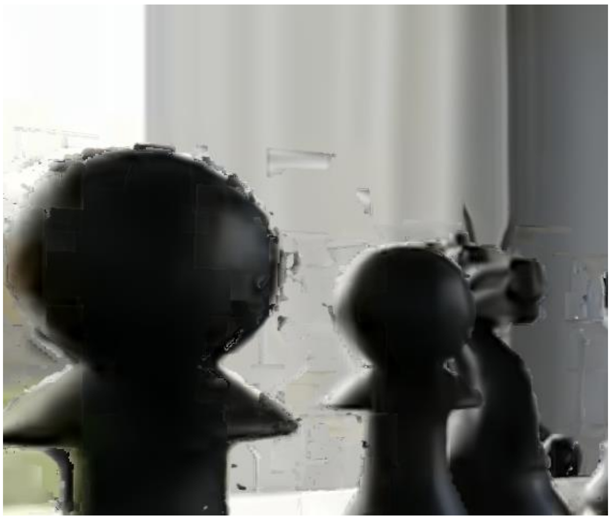
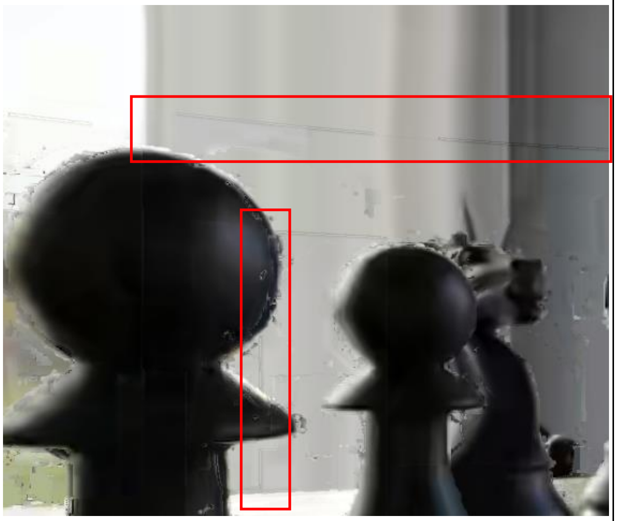
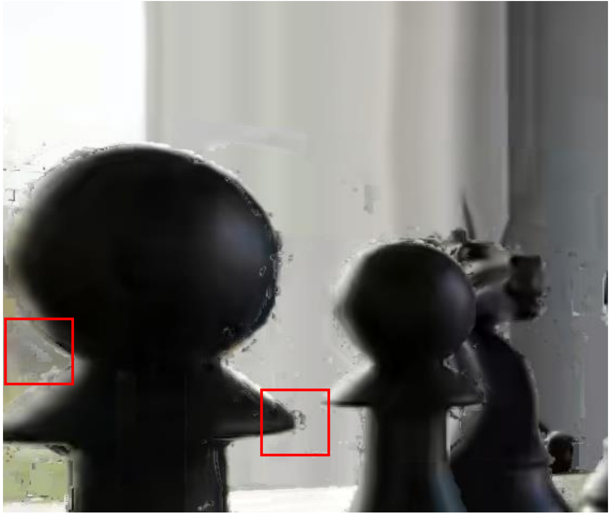
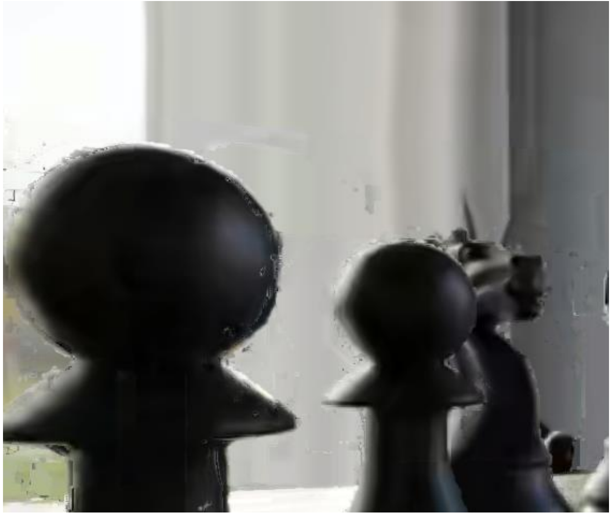
This operation allows reducing artifacts caused by blurred depth maps (thus is especially effective for lower bitrates with destroyed depth edges) visible as small disturbing artifacts close to edges of objects.

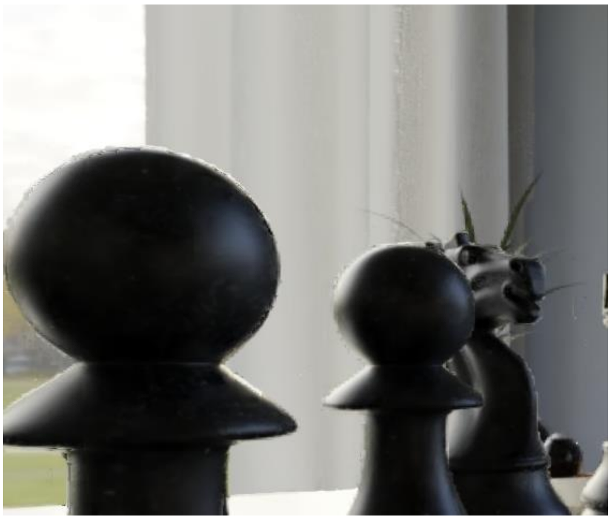
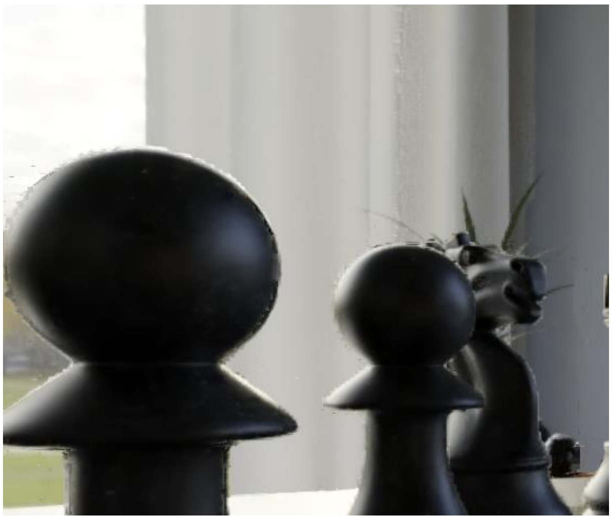
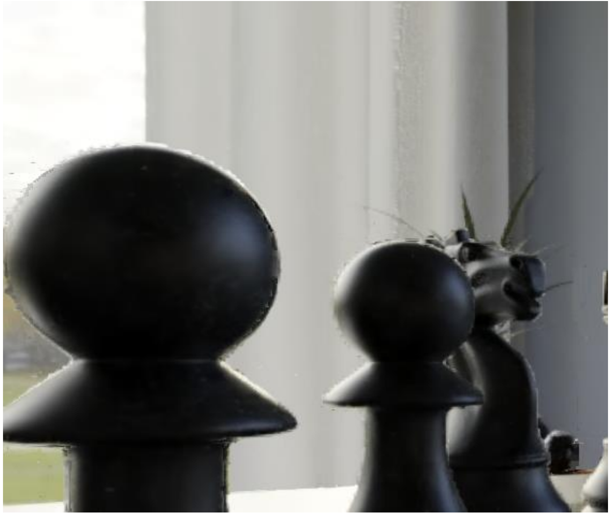
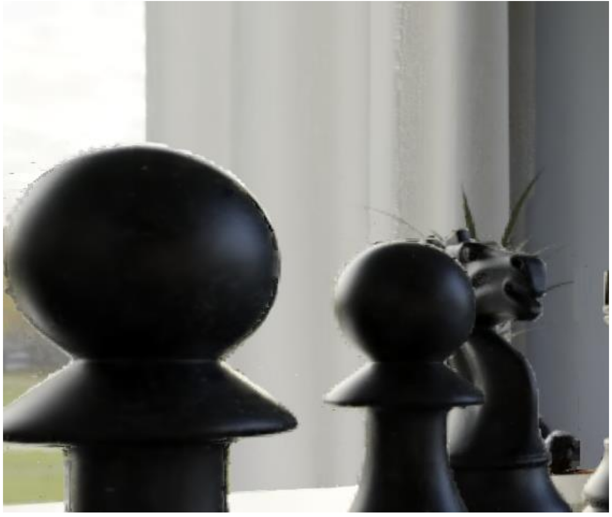
The filtering is performed within the frame rendering process:

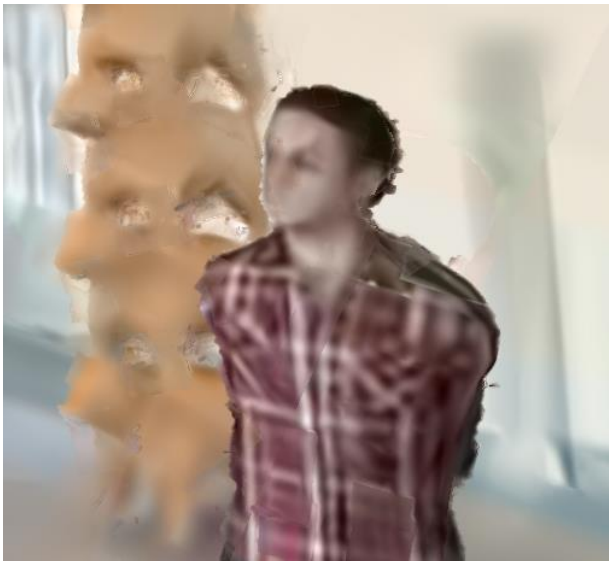
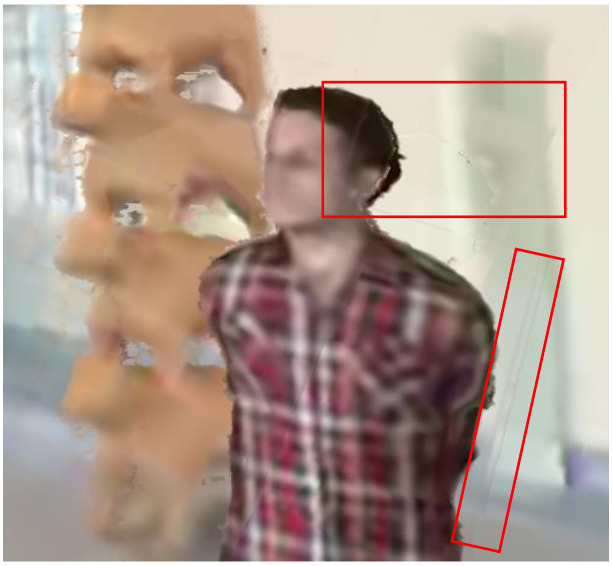
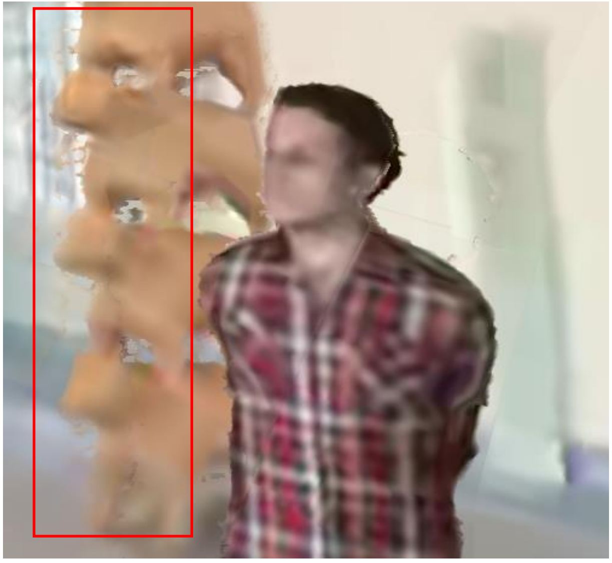
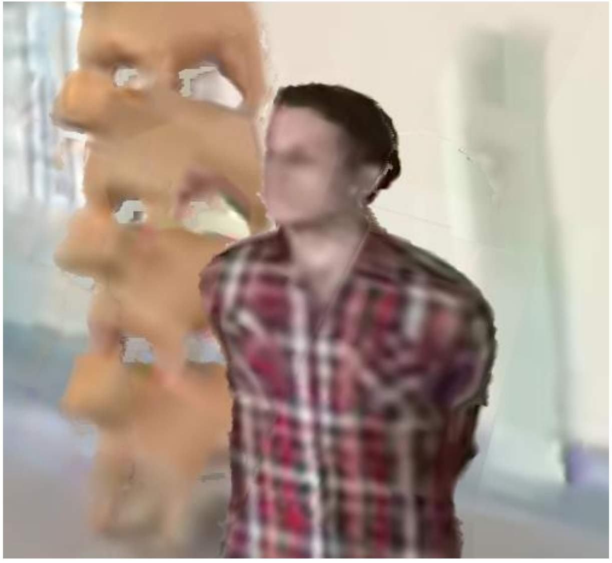


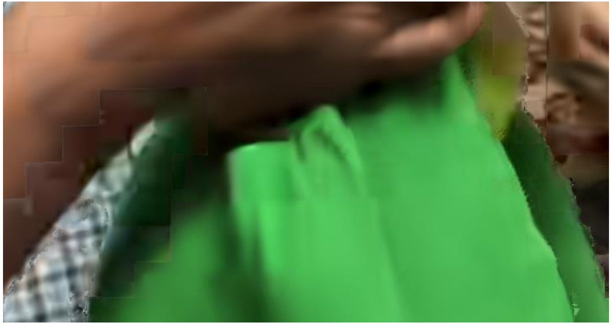
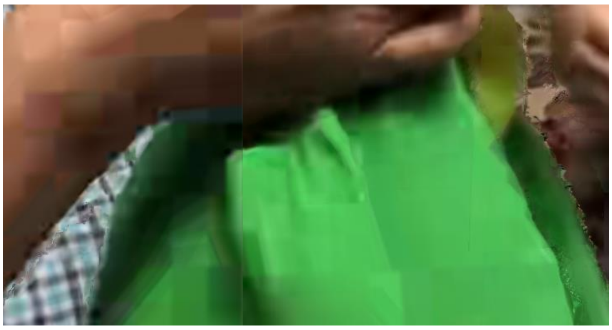
```
auto renderFrame(const MivBitstream::AccessUnit &frame,
                 const MivBitstream::CameraConfig &cameraConfig) -> Common::RendererFrame {
  const auto &viewParamsList = frame.viewParamsList;
  const auto sourceHelperList = ProjectionHelperList{viewParamsList};
  const auto targetHelper = ProjectionHelper{cameraConfig.viewParams};

  // 0) Initialization
  findInpaintedView(frame);
  computeCameraWeight(sourceHelperList, targetHelper);
  computeCameraVisibility(sourceHelperList, targetHelper);
  computeAngularDistortionPerSource(sourceHelperList);

  // 1) Deconstruction
  recoverPrunedSource(frame, sourceHelperList);

  // 2) Reprojection
  reprojectPrunedSource(frame, sourceHelperList, targetHelper);

  // 3) Warping
  warpPrunedSource(frame, targetHelper);

  // 3.5) Morphological filtration
  filterReprojectedPrunedDepthMaps(frame);

  // 4) Weight recovery
  recoverPrunedWeight(sourceHelperList, targetHelper);

  // 5) Selection
  selectViewportDepth(!frame.casps->casps_miv_extension().casme_depth_low_quality_flag(),
                      targetHelper);

  // 6) Filtering
  filterVisibilityMap();

  // 7) Shading
  computeShadingMap(sourceHelperList, targetHelper);

  // 8) Output
```

2 Results

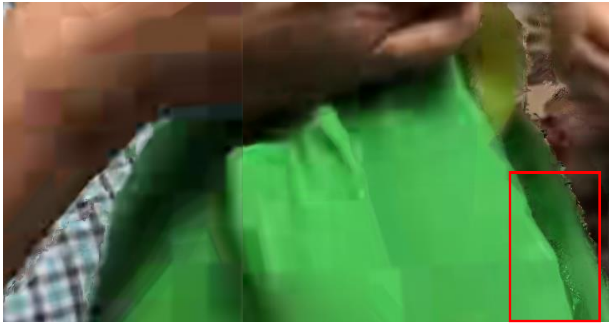
A97 anchor, SN, p01, QP5, f52 	MinPatchSize = 32 
MinPatchSize = 32, PatchMargin = 4 	MinPatchSize = 32, PatchMargin = 4, filtration 

A97 anchor, SN, p01, QP1, f52 	MinPatchSize = 32 
MinPatchSize = 32, PatchMargin = 4 	MinPatchSize = 32, PatchMargin = 4, filtration 

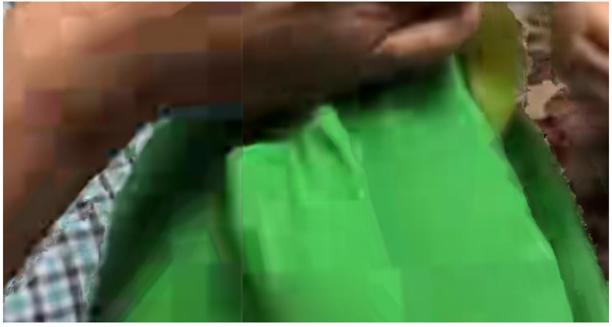
A97 anchor, SB, p01, QP5, f52 	MinPatchSize = 32 
MinPatchSize = 32, PatchMargin = 4 	MinPatchSize = 32, PatchMargin = 4, filtration 

A97 anchor, SE, p01, QP5, f52 	MinPatchSize = 32 

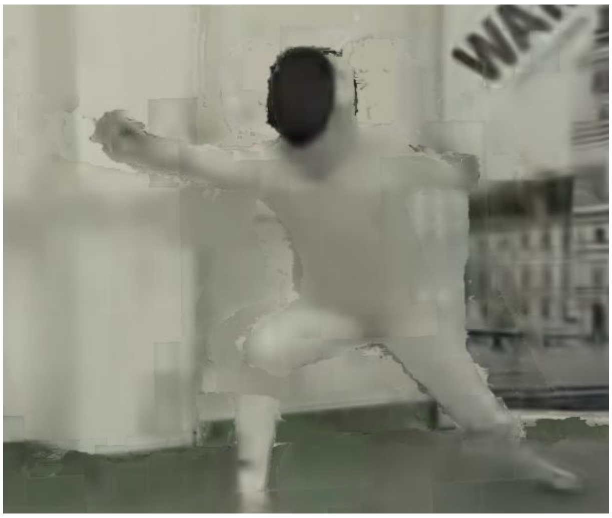
MinPatchSize = 32, PatchMargin = 4



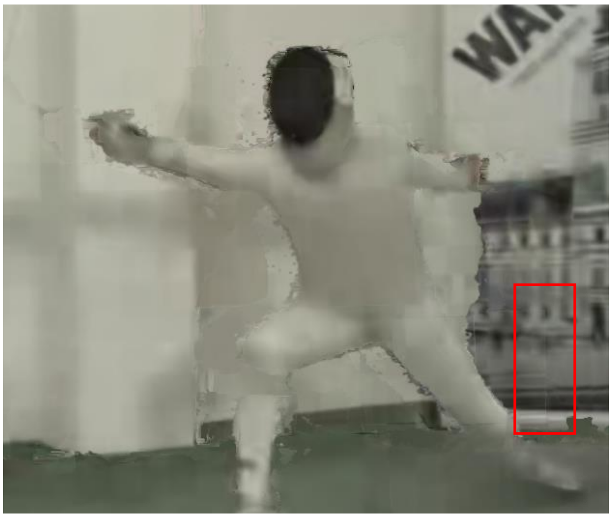
MinPatchSize = 32, PatchMargin = 4, filtration



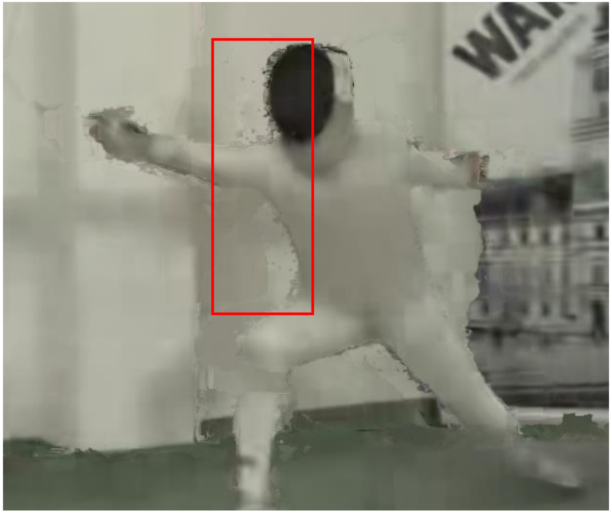
A97 anchor, SL, p01, QP5, f52



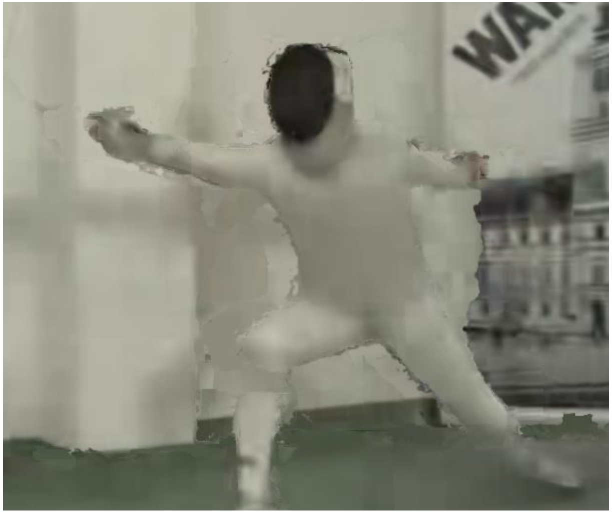
MinPatchSize = 32



MinPatchSize = 32, PatchMargin = 4



MinPatchSize = 32, PatchMargin = 4, filtration



3 Recommendation

We recommend:

- to watch provided posetraces,
- to include proposed renderer modifications (patch margins and morphological filtering) into TMIV13,
- to change the default min patch size value for TMIV anchor generation.

4 Acknowledgement

This work was supported by the Ministry of Science and Higher Education of Republic of Poland.