Problem and Motivation

- CGI Industry lately favors global-illumination (GI) and higher frame-rates, both of them increase visual quality but take longer for the image sequences to render.
- Temporal coherence (upsampling) methods for GI algorithms increase performance compared to regular rendering.
- Image-based upsampling techniques generate frames based on ground-truth ones and are in general significantly faster than the previous mentioned ones.
- However, upsampling techniques can not handle reflections and refractions correctly, because they result in motion of image features which is incoherent to the objects motion in the scene.

Contribution

- Two image-based upsampling techniques, that use temporal coherence to create additional frames.
- Independence from rendering algorithm allows for portability.
- Preserves image features such as specular or transparent surfaces better than previous work.
- First to utilize the non-local means denoising algorithm for upsampling frames, because it does not rely on object movement, in contrast to previous work, and it can be optimized for parallel execution.
- More accurate than some image-based previous work.

Motion-Vector Technique

- Take 2 B-frames (fully rendered ones).
- Find similar pixels from one frame to the other based on the color, using the non-local means algorithm (with the difference in color as $\xi$). Those similarities can be interpreted as motion vectors.
- For each pixel in the I-frame (the interpolated one) find the source pixel in the B-frames based on the motion vectors.
- Mix the source pixel together or use the scene-assisted method of Yang et al. [1] as fallback.

Pixel-Similarity Technique

- Render a stubby frame (only surface data) of the I-frame.
- Search in adjacent B-frames for points with similar surface data, using the non-local means algorithm.
- The similarity measure $E$ is made up of:
  - the diffuse surface similarity: point-location.
  - the specular surface similarity: location after reflection.
  - the transparent surface similarity: location after refraction.
- Color the I-frame based on the similarities.

Adapted Non-Local Means Algorithm

- Searches for similar pixels $b$ in a pre-defined neighbourhood $H$ around $a$.
- Two pixels $a$ and $b$ are similar if the pixels in their surrounding $S$ match.
- A match in the surrounding is calculated by the similarity measure $\xi$.
- How similar two pixels are, is defined by the sum of the similarity measures results.
- Can be optimized for parallel execution.
- Adaption stores the $n$ most similar pixels.

Results

- Handles transparent objects better than other methods.
- Not data-dependent, unlike pixel-similarity technique.
- Color bleeding when new geometry shows, for example due to rotation.
- Edge location is off due to non-linear movement.
- Handles reflections better than other methods.
- Edge-location always similar to ground-truth.

Variable $H$ is the pixel neighbourhood around the point $a$. Variable $S$ is the pixel surface neighbourhood. The similarity measure $\xi$ is made up of:

$$\xi(a, b) = \sum_{h \in H} \sum_{s \in S} \xi(a, s, a+h, s)$$

Unoptimized adapted non-local means:

```python
for each pixel a in image do
  for each h in neighbourhood H do
    E = 0;
    for each s sourrounding S do
      calculate E += \xi(a+s, a+h+s)
    end keep if E maximal
  end
```

Refactored version:

```python
for each pixel a in image do
  for each h in neighbourhood H do
    E = 0;
    for each s sourrounding S do
      calculate E += \xi(a+s, a+h+s)
    end keep if E maximal
  end
```

References: