# SIGGRAPH 2012

The **39th** International **Conference** and **Exhibition** on **Computer Graphics** and **Interactive Techniques**

# Accelerating Rendering Pipelines Using Bidirectional Iterative Reprojection

Lei Yang
Bosch Research (Palo Alto, CA,USA)

Huw Bowles
Gobo Games (Brighton, UK)

Additional contributors:

Kenny Mitchell
Disney Research

Pedro Sander
Hong Kong UST

# Overview

SIGGRAPH2012

- Introduction
- Iterative reprojection
- Bidirectional reprojection
- Conclusion

# The papers

SIGGRAPH2012

- Two papers (concurrent work) on iterative reprojection:

- **Iterative Image Warping**
  *H. Bowles, K. Mitchell, B. Sumner, J. Moore, M. Gross*
  *Computer Graphics Forum 31(2) (Proc. Eurographics 2012)*

- **Image-space bidirectional scene reprojection**
  *L. Yang, Y.-C. Tse, P. Sander, J. Lawrence, D. Nehab, H. Hoppe, C. Wilkins.*
  *ACM Transactions on Graphics, 30(6) (Proc. SIGGRAPH Asia 2011)*

# Split/Second



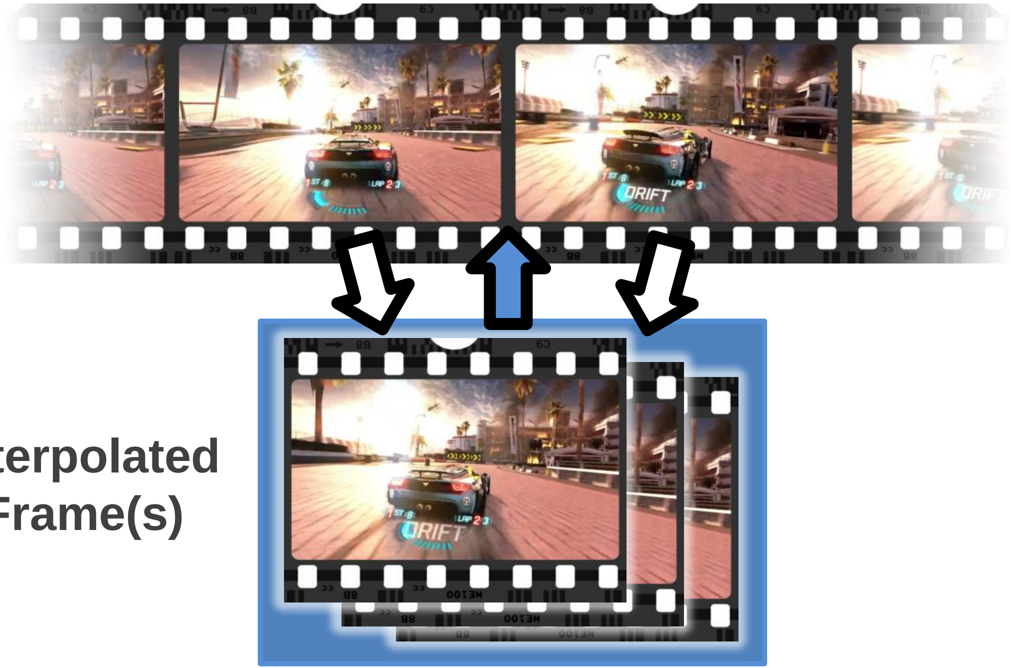
# Traditional pipelines

SIGGRAPH2012

- Current graphics architectures require brute force rendering of every frame, so they don't scale well to high frame rates
- However, nearby frames are usually very similar thanks to temporal coherence
- We can synthesize a plausible frame without performing the rasterization and shading, by reusing rendering results from neighbouring frame(s)

# Frame interpolation

Rendered Frames



Interpolated Frame(s)

# Real-time reprojection strategies



- Rasterize scene from target viewpoint and sample shading from the source viewpoints (Nehab2007)
- Warp the existing frames using per-pixel primitives into the target viewpoint (Mark1997)
- Use some kind of approximation (Andreev2010, Didyk2010)
- Warp frames using an iterative search (Yang2011, Bowles2012)
- See papers for detailed comparison

# Overview

SIGGRAPH2012

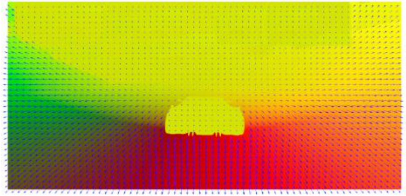
- Introduction
- Iterative reprojection
  - Algorithm
  - Iteration initialisation
  - Disocclusion handling
- Bidirectional reprojection
- Conclusion

# Iterative reprojection



Rendered Frame [t]


Motion Vectors



Target Frame [t+$\alpha$]

$$p_{tgt} = p_{src} + V(p_{src})$$

# Image-based iterative reprojection

SIGGRAPH2012

- Know mapping of each pixel via equation:

$$p_{tgt} = p_{src} + V(p_{src})$$

- Run a GPU shader over the target frame: $p_{tgt}$ known

- Problem: How to solve for $p_{src}$?

# Iterative solution



- Know mapping of each pixel via equation:

$$p_{tgt} = p_{src} + V(p_{src})$$

- Idea - Solve iteratively:

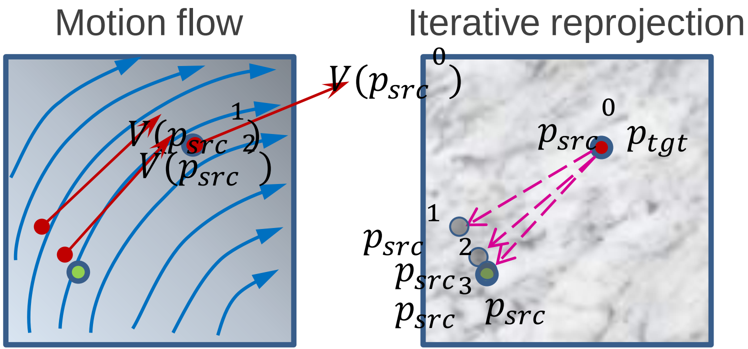
$$p_{src}{}^{i+1} = p_{tgt} - V(p_{src}{}^{i})$$

- Fixed Point Iteration

# Iterative solution



- Algorithm
  1. Pick a start point: ${p_{src}}^0$ (e.g. $p_{tgt}$)
  2. Apply recurrence relation until convergence: ${p_{src}}^{i+1} = p_{tgt} - V({p_{src}}^i)$



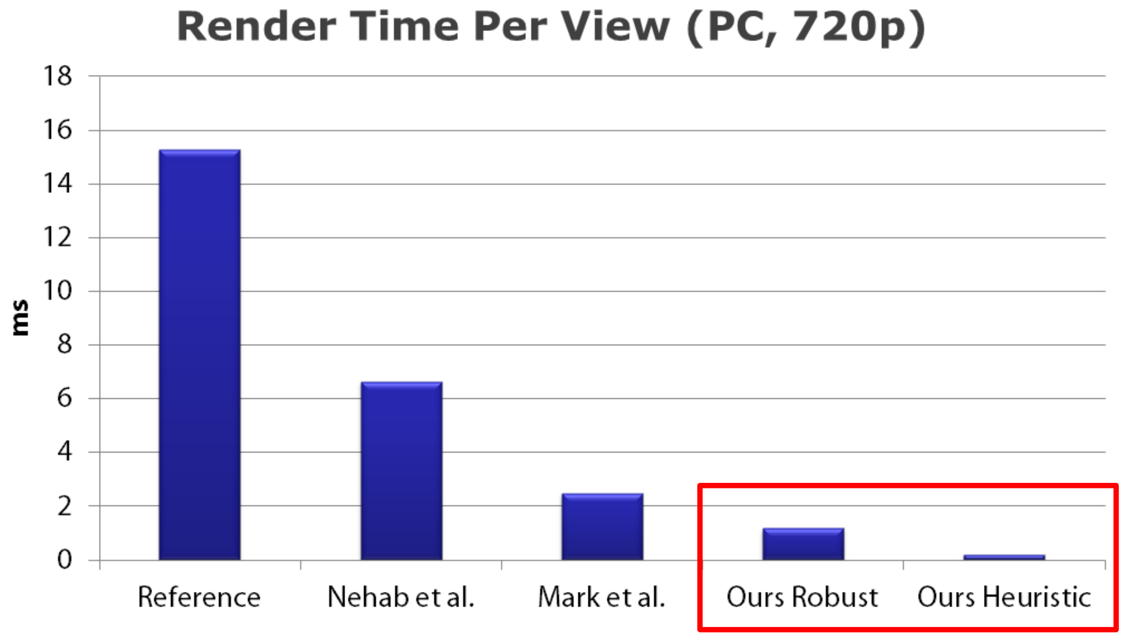
# Single frame reprojection – Split/Second scene (6x slow motion)

60Hz (With reproj. frames)

30Hz (Original)

# Performance

SIGGRAPH2012

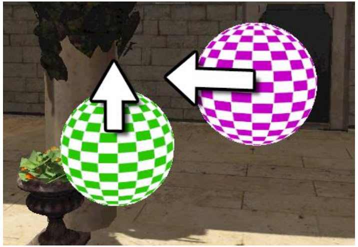


# Considerations



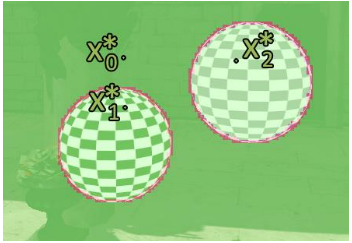
- Iteration initialisation
- Disocclusions

# Iteration initialisation

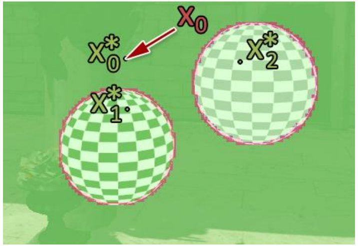


Source

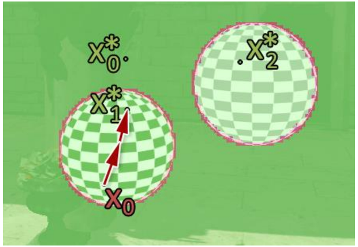

Target



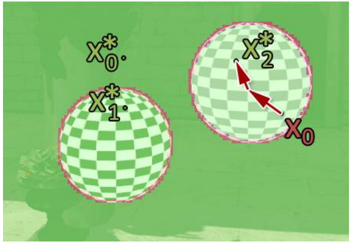

Source Analysis




Background



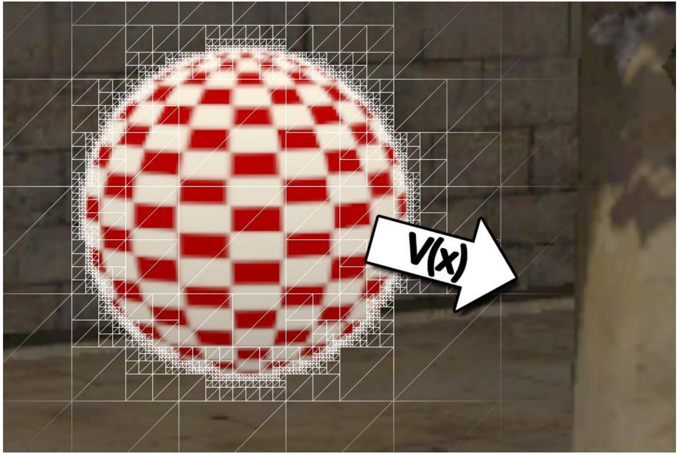

Green Sphere

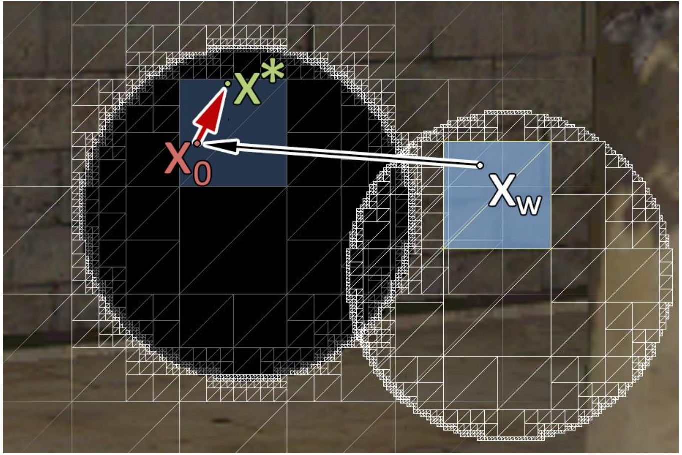



Purple Sphere

# Iteration initialisation

- Subdivide into quads and rasterize at warped positions (Bowles2012)



# Disocclusions



# Disocclusions



# Disocclusions



- Reshading (Nehab2007)
  - Requires traversing the scene again
- Inpainting (Andreev2010, Bowles2012)
  - Image-based
  - Depends on the hole size and visual saliency of the region
- Bidirectional reprojection (Yang2011)

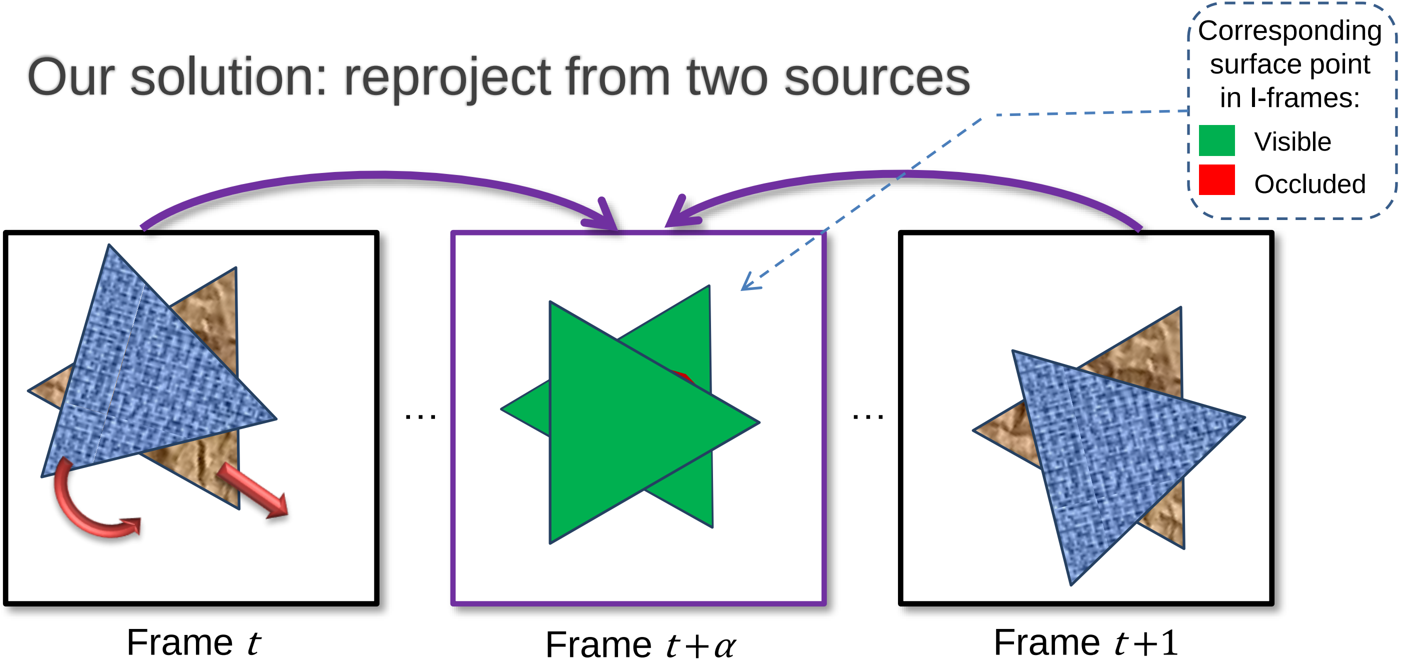
# Overview

SIGGRAPH2012

- Introduction
- Iterative reprojection
- Bidirectional reprojection
  - Algorithm
  - Practical details
  - Results
- Conclusion

# Reducing disocclusion

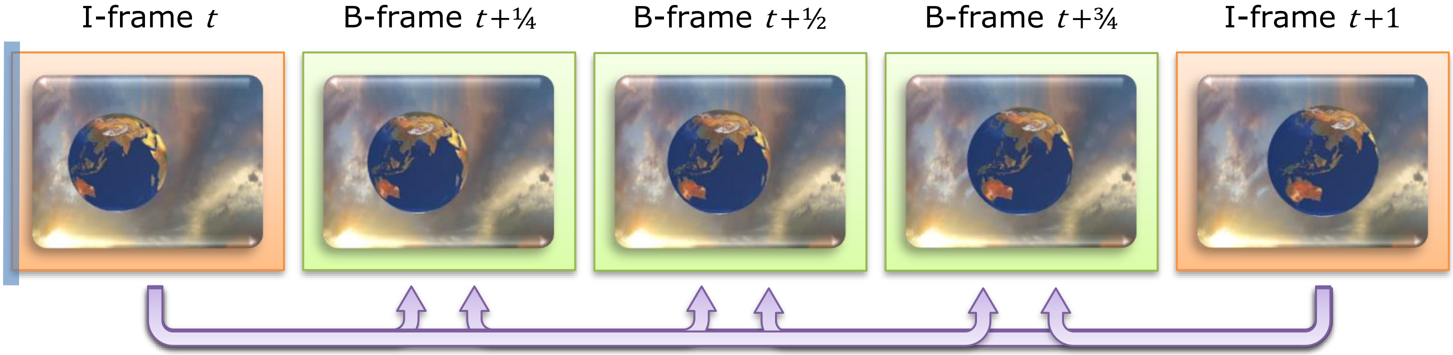
- Our solution: reproject from two sources



# Bidirectional reprojection



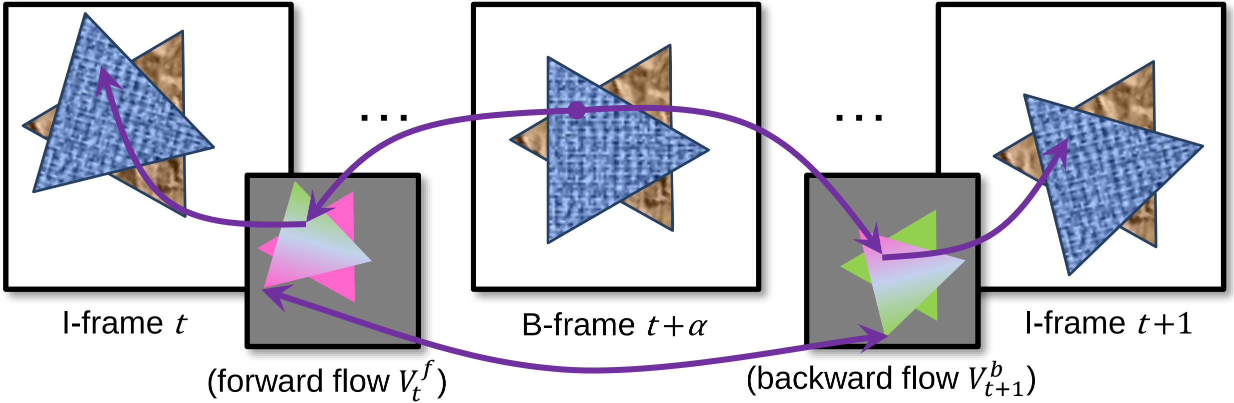
- Scenario: frame **interpolation**:
  Render *I-frames* *(Intra-frames, or key-frames)*,
  Insert interpolated *B-frames* *(Bidirectionally interpolated-frames)*
- *"Bidirectional Reprojection"* (*Bireproj*)



# Bidirectional reprojection



- Generate motion flow fields for each pair of I-frames
- For each pixel in B-frame $t+\alpha$
  - Search in forward flow field $V_t^f$ to reproject to I-frame $t$
  - Search in backward flow field $V_{t+1}^b$ to reproject to I-frame $t+1$
  - Load and blend colors from frame $t$ and $t+1$

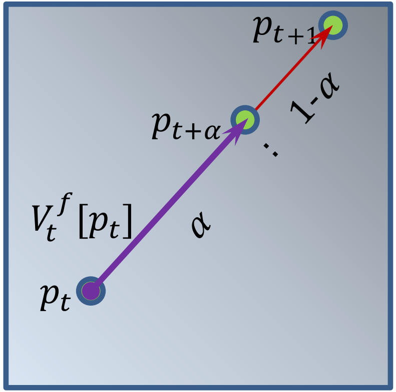


# Iterative reprojection

SIGGRAPH2012

- Motion flow fields map pixels between I-frames $t$ and $t+1$
  - Independent of $\alpha$
- Assume the motion between $t$ and $t+1$ is linear: scale the vectors by $\alpha$ (or $1-\alpha$ )
- Use iterative reprojection to solve $p_{t+\alpha}$

Motion flow field



# Motion vector field generation

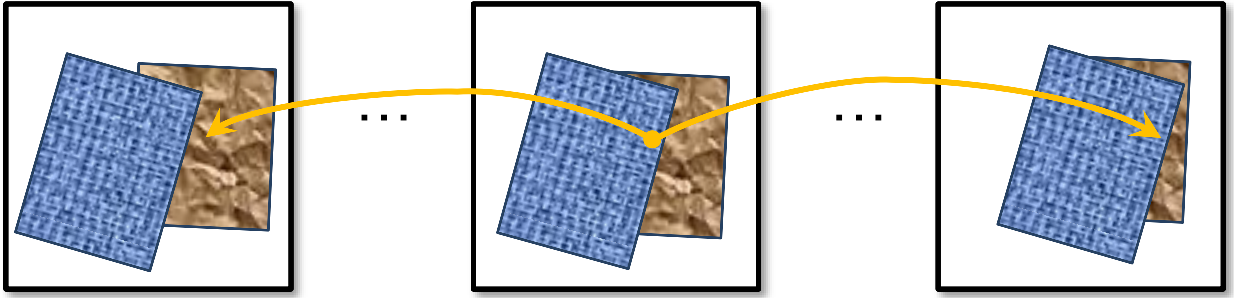


- Additional position transform in the VS
- $V^b$ commonly found in the G-buffer (for motion blur)
- Missing forward motion field $V_t^f$?
  - Negate the field $V_t^b$
  - Use iterative reprojection to improve the precision (based on a precise $V_{t+1}^b$)

# Choosing the right pixel



- The results from frame $t$ and $t+1$ may disagree
- Reasons:
  - Occlusion: one source is occluded by the other in $t+\alpha$
    - *choose the visible one based on the interpolated depth*

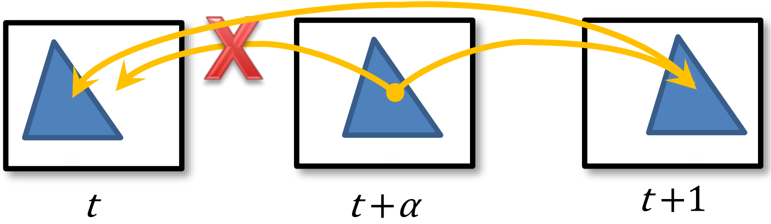


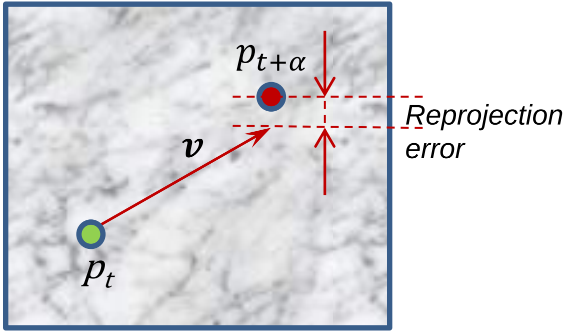

I-frame $t$ B-frame $t+\alpha$ I-frame $t+1$

# Choosing the right pixel



- The results from frame $t$ and $t+1$ may disagree
- Reasons:
  - Incorrect reprojection: iterative reprojection failed
    - *Sign: reprojection error -- residual between $p_t + \boldsymbol{v}$ and $p_{t+\alpha}$*
    - *mutual correction between $p_t$ & $p_{t+1}$ with correspondence*

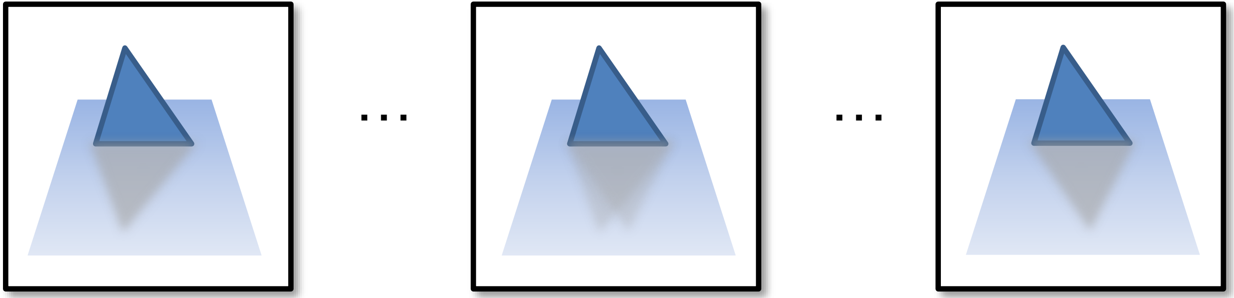


# Choosing the right pixel



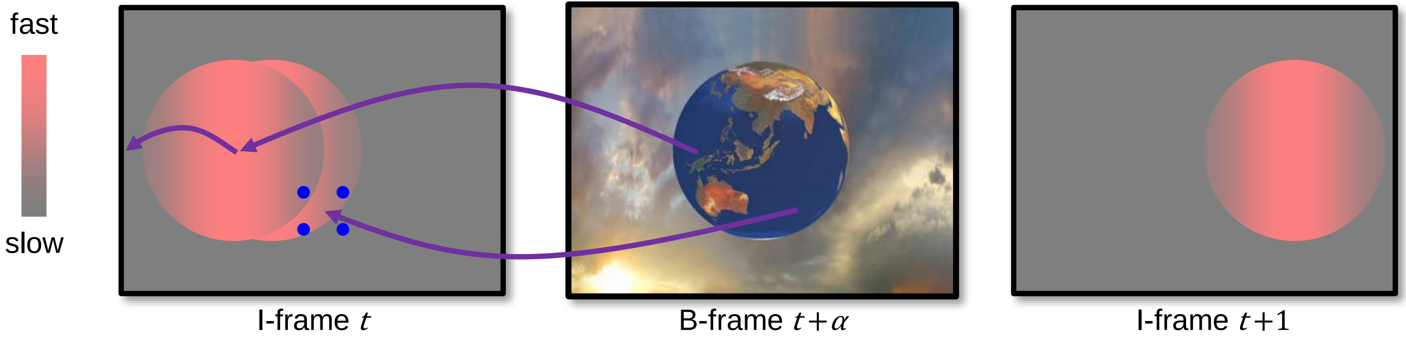
- The results from frame $t$ and $t+1$ may disagree
- Reasons:
  - Shading changed: lighting, shadows, dynamic texture…
    - interpolate the results based on $\alpha$




I-frame $t$ B-frame $t+\alpha$ I-frame $t+1$

# Additional search initialization

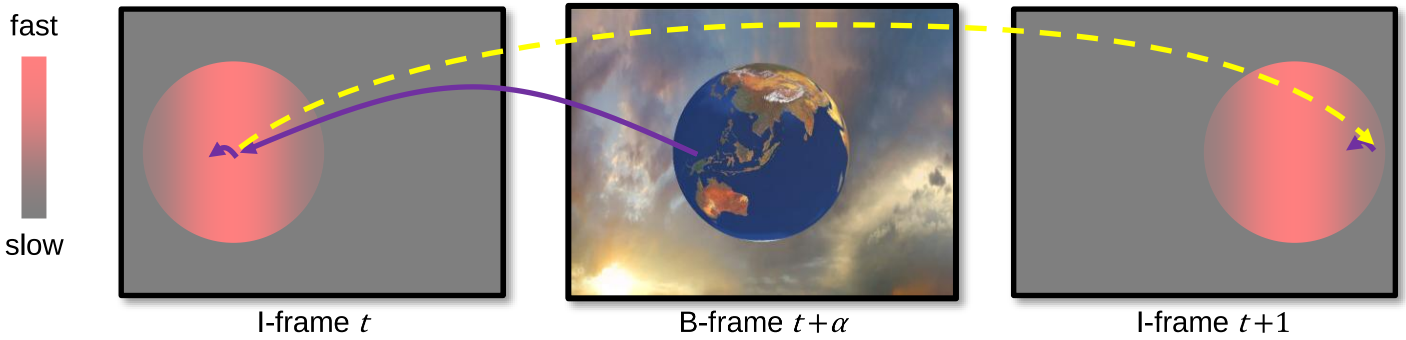
- Problems when using the target pixel as iteration starting point
  - a) Imprecise initial vector across object boundaries
  - b) Search steps can fall off the object
- For a) :
  - Additional 4 candidates within a small neighborhood
  - Initialize using the result from a closer B-frame



# Additional search initialization

SIGGRAPH2012

- The motion field is often only piecewise smooth
  - a) Imprecise initial vector across object boundaries
  - b) Search steps can fall off the object
- For b):
  - Initialize using the vector from the opposite I-frame



# Additional search initialization

I-frame $t$

…

B-frame $t+½$

…

I-frame $t+1$

Image-based
(No additional init.)

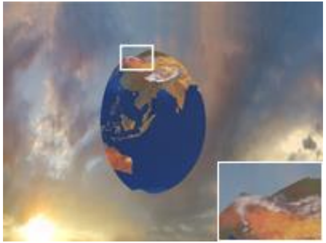


Image-based
(with “b”)

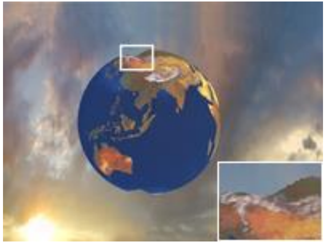
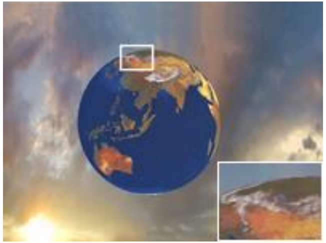


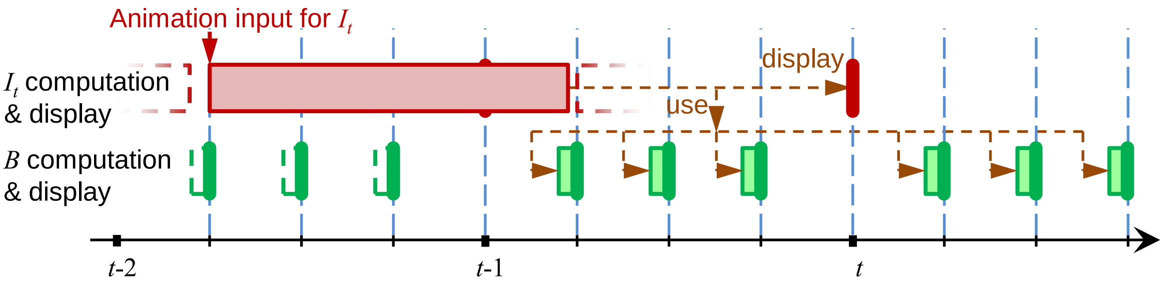
Image-based
(with “a+b”)



# Partitioned rendering



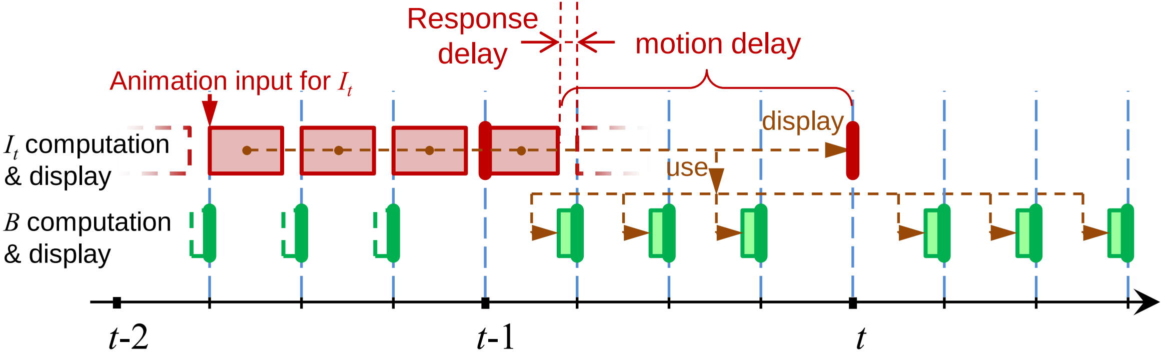
- I-frame shading parallel to B-frame generation
- Partition the I-frame rendering tasks evenly
  - Straightforward for games that has hundreds or more draw calls per frame
  - Runtime: interleave B-frame generation (green) with I-frame rendering (red)
  - Possible: no need to partition with (future) GPU multitasking



# Lag



- I-frame "$t$" must start rendering at $t-1-\frac{n-1}{n}$ ($n$=4 here)
  - Introduces a potential lag to the pipeline – I-frame delayed by $\frac{n-1}{n}$
  - However: the motion of frame $t$ is already seen at B-frame $t-\frac{n-1}{n}$



# Lag

SIGGRAPH2012

- Lag with standard double buffering:
  - Original: 1 *time step (ts)*
  - Bireproj: *position:* $1 + \frac{n-1}{n}$ *ts*, *response:* 1 *ts*
- Lag with 1-frame render ahead queue:
  - Original: 2 *ts*
  - Bireproj: 2 *ts (position)*
- Theoretical / empirical analysis (Yang2011)

# Bireproj results



- Example: three B-frames per I-frame time step
- 2-3ms for a B-frame (1280x720)
- Suitable scenarios:
  - Vertex-bound scenes
  - Fill-bound scenes
  - Multi-pass / deferred rendering

# Bireproj results – the *walking* scene

- Fill-bound scene with an expensive pixel shader (2.6x speed-up)

# Bireproj results – the *terrain* scene

SIGGRAPH2012

- Geometry bound scene (1M triangles) (2.8x speed-up)

# Bireproj results – the *head* scene

- Multi-pass skin rendering [d'Eon and Luebke 2007] (2.6x speed-up)

# Bireproj results – shading interpolation

SIGGRAPH2012

- Reduce popping artifacts with dynamic lighting and shadows

# Bireproj results – Split/Second

SIGGRAPH2012

- Results from ***Split/Second*** by Black Rock Studio
  - Input: an image set with corresponding depth and backward motion vector fields
  - Some of the edge artifacts are caused by imprecise depth
  - A stress test for Bireproj

# Split/Second Airport Scene (4x Slow Motion)

Original

Bireproj

# Split/Second Airport Scene
# (4x Slow Motion)

Original

Bireproj

# Limitations



- Dynamic shading interpolation
  - ☹ Does not work when visible in only one source
  - ✓ Separate and render the problematic components per B-frame
- Fast moving thin object visibility
  - ☹ Reprojection may be improperly initialized
  - ✓ Use robust initialization (with DX 10+ level hardware)
- Bireproj introduces a small lag
  - ☹ Less than one (I-frame) timestep of positional delay
  - ✓ Response delay is minimum (≈0)

# Summary

SIGGRAPH2012

- Reuse shading results to reduce redundant computation
- Image-based iterative reprojection
  - Purely image-based (no need to traverse the scene)
  - Fast – 0.85 ms on PS3 (1280x720)
  - Very accurate reprojection when given proper initialization
- Bidirectional reprojection
  - Almost eliminates disocclusion artifacts
  - Boosts framerate by almost *n* (# of interpolated frames) times
  - Interpolates dynamic shading changes

# Further details



- Refer to [Bowles et al 2012] for:
  - Application to general image warps, inc. spatial rerpojections and non-linear temporal reprojection
  - Analysis of convergence properties of FPI
  - Robust initialization algorithm
- Refer to [Yang et al 2011] for:
  - Bireproj using traditional reverse reprojection
  - Hybrid geometry/image-based reprojection
  - Theoretical & empirical lag analysis

# Thank you!


- Acknowledgements
  - Paper authors group 1 (IIW): *K. Mitchell, B. Sumner, J. Moore, M. Gross*
  - Paper authors group 2 (Bireproj):*Y.-C. Tse, P. Sander, J. Lawrence, D. Nehab, H. Hoppe, C. Wilkins.*
  - Disney Interactive Studios (for the Split/Second assets)
  - NVIDIA and XYZRGB (for the human head assets)

# References


- Mark W. R., McMillan L., Bishop G. “*Post-rendering 3D Warping*”, I3D 1997
- Nehab D., Sander P., Lawrence J., Tatarchuk N., Isidoro J. “*Accelerating real-time shading with reverse reprojection caching*”, Graphics Hardware 2007
- Andreev D., “*Real-time frame rate up-conversion for video games*”, SIGGRAPH Talk 2010
- Bowles H., Mitchell K., Sumner R. W., Moore J., Gross M., “*Iterative Image Warping*”, Eurographics 2012
- Yang L., Tse Y.-C., Sander P. V., Lawrence J., Nehab D., Hoppe H., Wilkins C. L. “Image-based bidirectional scene reprojection”, SIGGRAPH Asia 2011
- Didyk P., Eisemann E., Ritschel T., Myszkowski K., Seidel H.-P., “Perceptually-motivated Real-time Temporal Upsampling of 3D Content for High-refresh-rate Displays”, Eurographics 2011