**Motivation**

- The traditional two-step view morphing can not run at real time.
- Morphing two images at a time.
- Morphing the in-between image with the third image.
- Other image-based rendering techniques are not able to navigate at an interactive rate due to expensive computations.
  - Light field and Lumigraph.
  - Uncalibrated point transfer techniques.
- Real-time tri-view morphing
  - Extend traditional two-step view-morphing to tri-view morphing based on epipolar constraint.
  - Cope well with both complex outdoor scenes and wide baseline images.

![Motivation Diagram](image)

A triple of images are the minimum unit for our algorithm. Our method is implemented as three steps: pre-warping, morphing, and post-warping.

**Algorithm Overview**

- **Two-step view morphing:**
  \[
  I_i = (1-s)H_i I_2 + s H_i I_1 \\
  I_{mix} = (1-t)H_i I_2 + t H_i I_1 \\
  I_{mix} = (1-s)H_i I_2 + s H_i I_1 + (1-t)H_i I_1 + t H_i I_2
  \]

- **Real-time tri-view morphing:**
  In the case of two image planes are parallel to each other, the epipolar lines of two images are horizontal. Following parallel image epipolar constraint, we obtain:
  \[
  \begin{align*}
  p^T F p_i &= 0 \\
  p^T F p_i &= p^T F(p_i + (s_i - s_j)(D_{mix}(p_i) - p_j)) \\
  &= (s_i - s_j)p^T F(s) - 0 = 0
  \end{align*}
  \]

So \( I_{mix} \) can be written as
\[
I_{mix} = (1-s)\hat{I}_2 + s \hat{I}_1 + (1-t)\hat{I}_1 + t \hat{I}_2
\]

Where
\[
\begin{align*}
\hat{I}_1 &= H_2 H_1 I_1 \\
\hat{I}_2 &= H_2 D_{mix}(H_1 I_1) \\
\hat{I}_1 &= H_2 H_1 D_{mix}(H_1 I_1)
\end{align*}
\]

- **Post-warping:**
  We extend a two-image post-warp algorithm to work with three images and yield the normal view.

**Experimental Results**

- Results of wide baseline images. The first row are three wide baseline sample images. The second row are a series of synthesized virtual views.
- Results of images taken with a hand-held digital camera. The first row are four uncalibrated sample images. The second row are a series of synthesized virtual views.
- Results of automatic transition between triples to create a long smooth walkthrough.

![Experimental Results](image)