Data-driven Hallucination of Different Times of Day from a Single Outdoor Photo

YiChang Shih[1], Sylvain Paris[2],
Frédo Durand[1], and William T. Freeman[1]

[1]MIT CSAIL [2]Adobe

Hallucinating scene color variation over time

- Goal: use the photo at time A to predict the photo at time B.

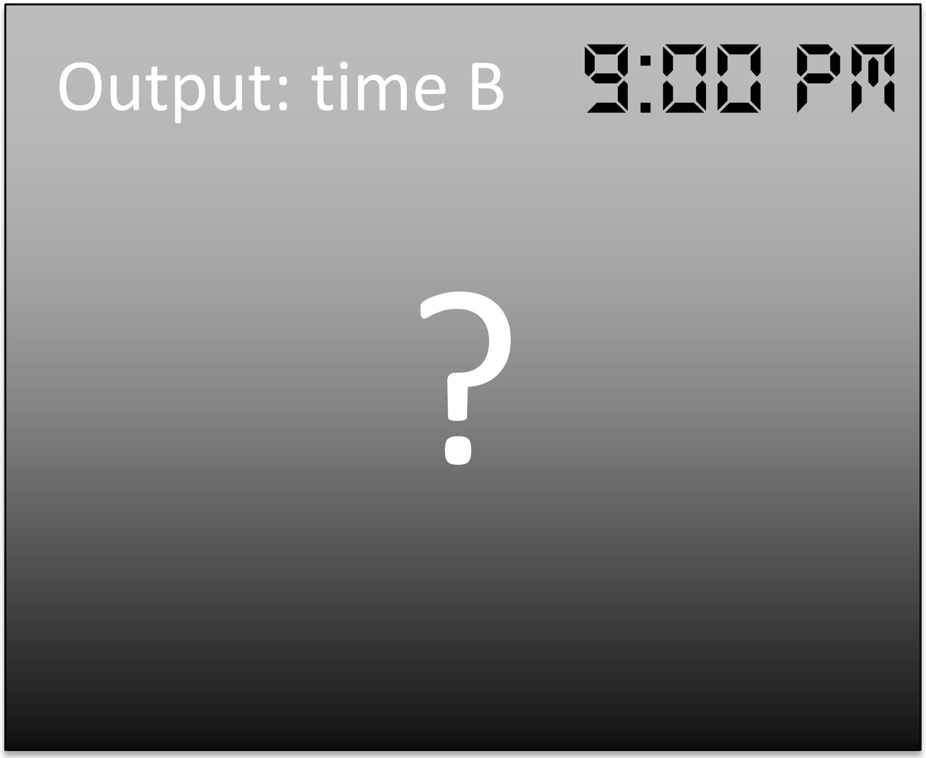


Image courtesy of Ken Chang

Hallucinating scene color variation over time

- Goal: use the photo at time A to predict the photo at time B.

Image courtesy of Ken Chang

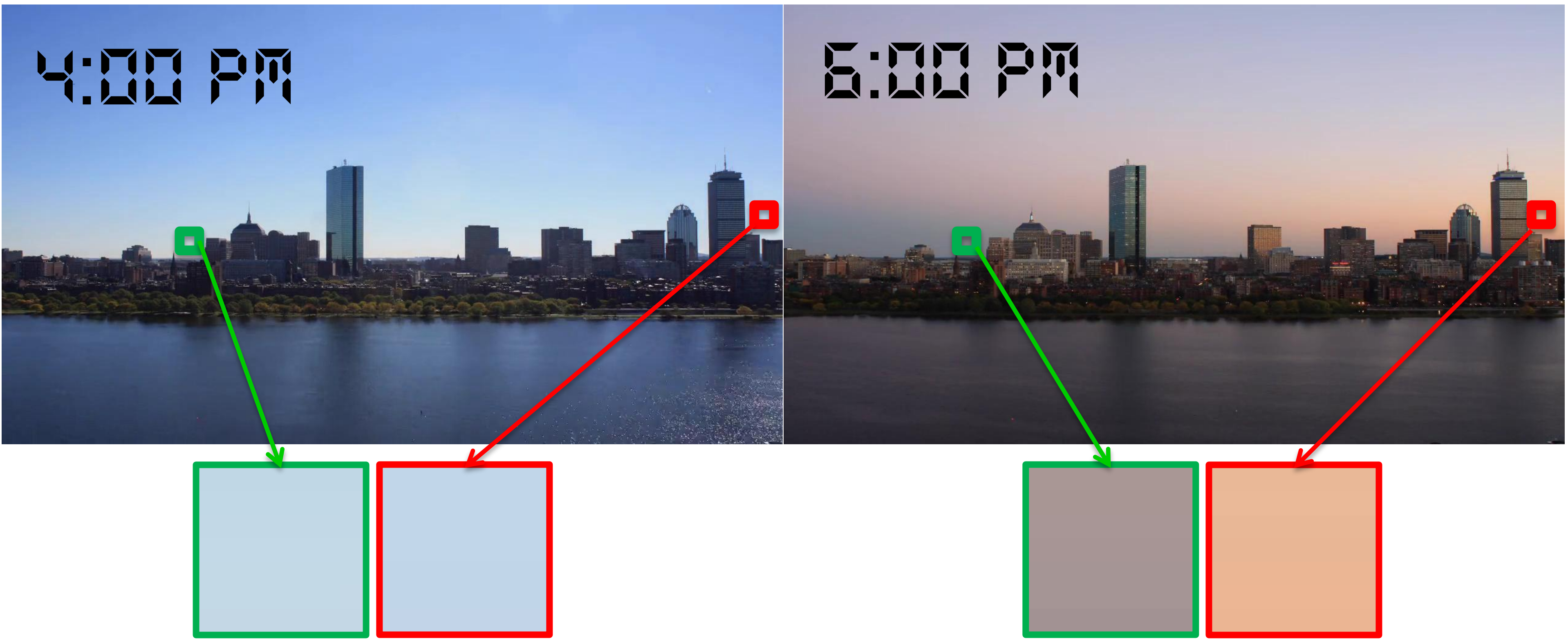
46 minutes too early [kenrockwell.com]

perfect [kenrockwell.com]

39 minutes too late [kenrockwell.com]

Hard problem

- The color change is spatially-variant!



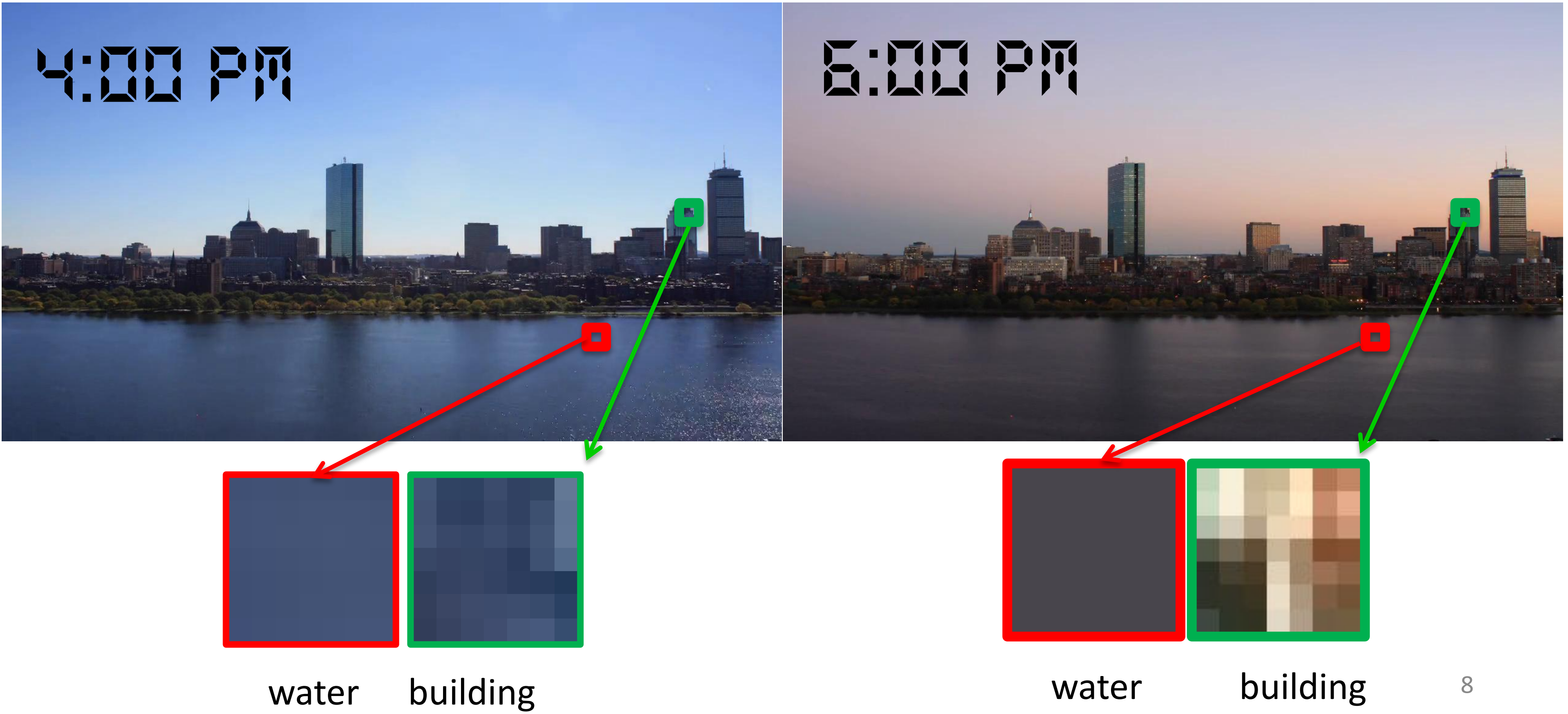
Colors are close at day

Become very different at sunset

Image courtesy of Adrian Dalca

Hard problem

- The color change depends on object
  - water and building become different color

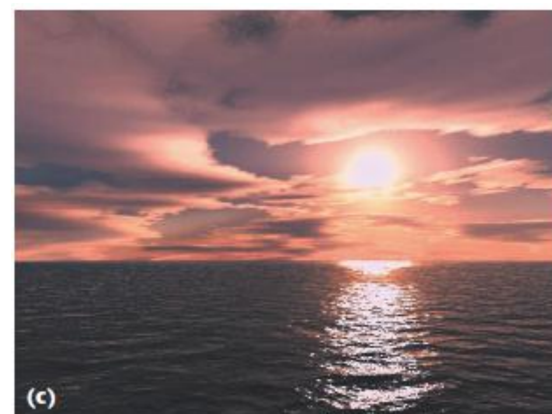


Related work: global color transfer

- Works on simple scenes [Reinhard et al, 2005] [Pouli and Reinhard, 2011] [Pitie et al. 2005]
- But complex scenes require spatially-variant color transfer

Input at daytime

Example at sunset

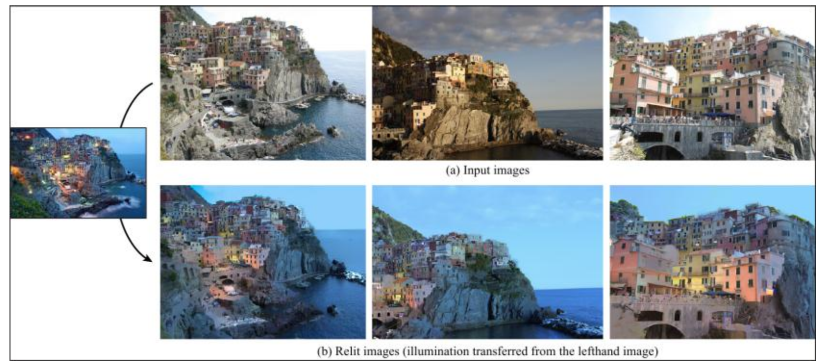


Output at sunset

Related work: image relighting

- Intrinsic images [Laffont et al., 2012]
  - need image collection of the scene

Inputs



Relit results

- Deep Photo [Kopf et al., 2009]
  - need 3D information

Input

Relit result

- We want a general machinery,
  not rely on data for specific input image

Related work: analyzing time-lapse sequence

- Produce good results, but need manually modeling the scene [Lalonde et al, 2009]

Input at daytime | Time A | Time B | Time C

Problem statement

- Input: single photo + target time of day
- Output: the same scene as if it was taken at the target time of day

Output at new time

?

- Requirement: fully automatic, no user input

Key idea: using time-lapse videos

Containing videos

- 500 videos at various scenes
- Labeled with time of day

Overview

Target time: 9pm

Overview

1. Match input to video from database

Matched time-lapse video

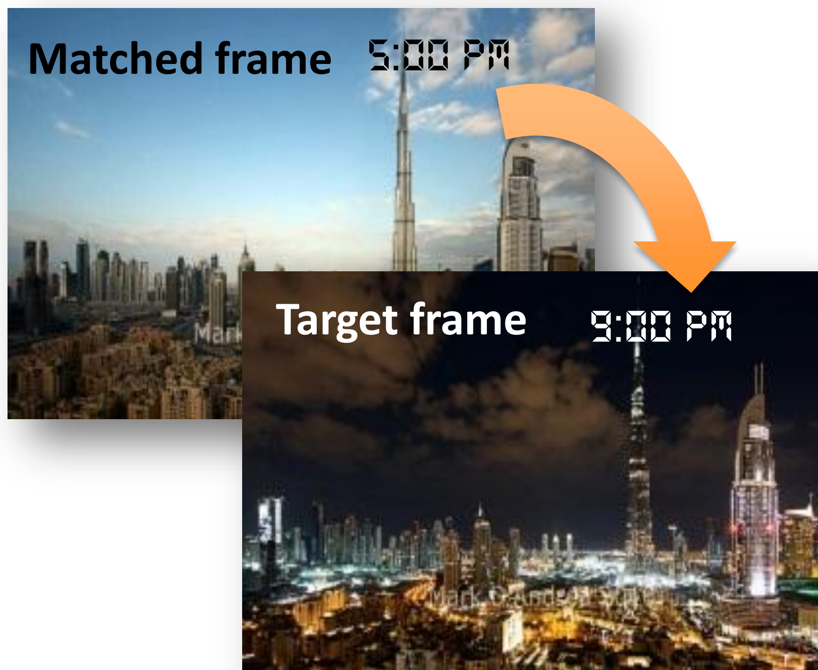
Target time: 9pm

Overview

1. Match input to video from database

Target time: 9pm

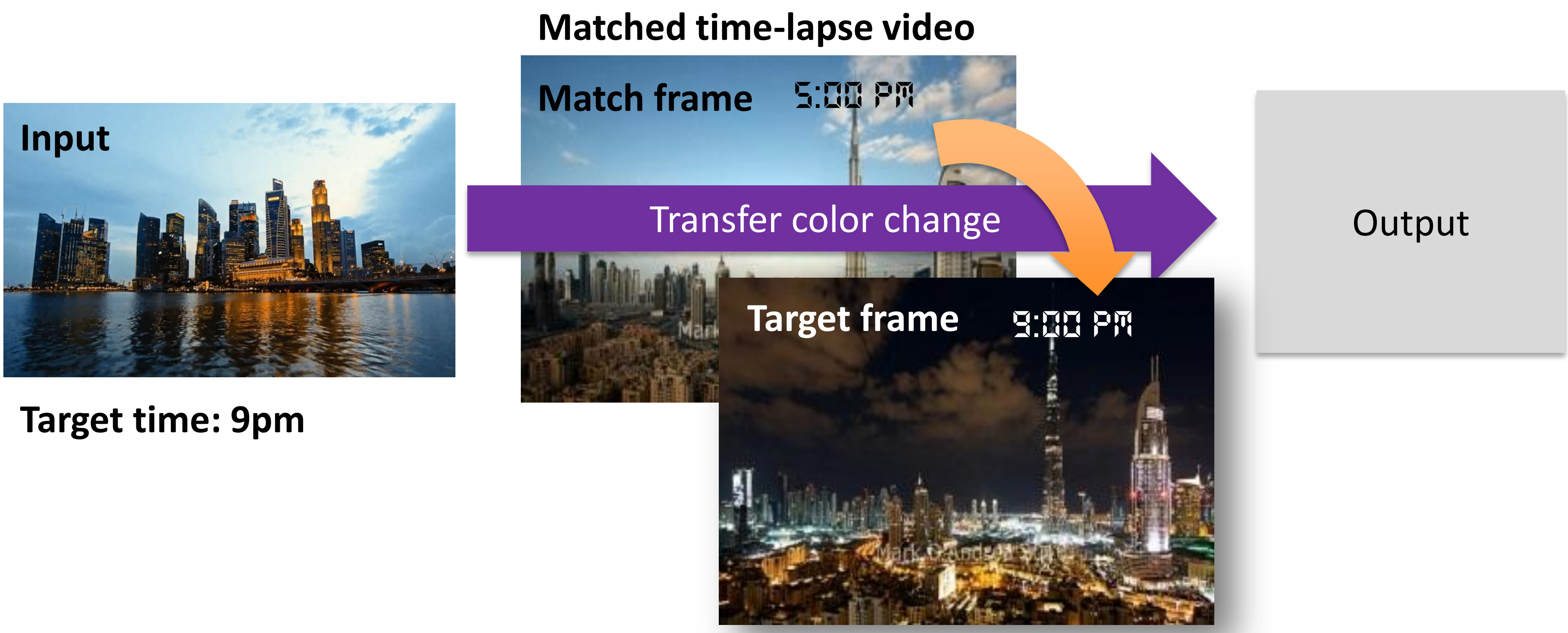
Matched time-lapse video



Overview

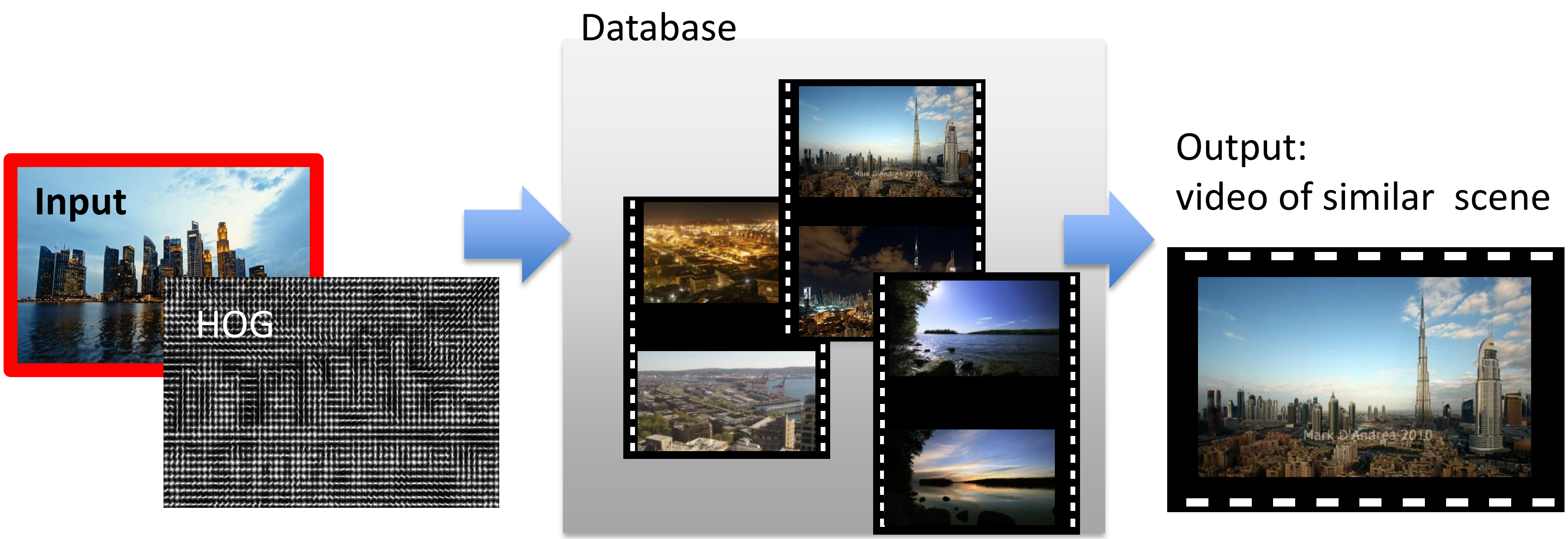
1. Match input to video from database

2. Transfer color change



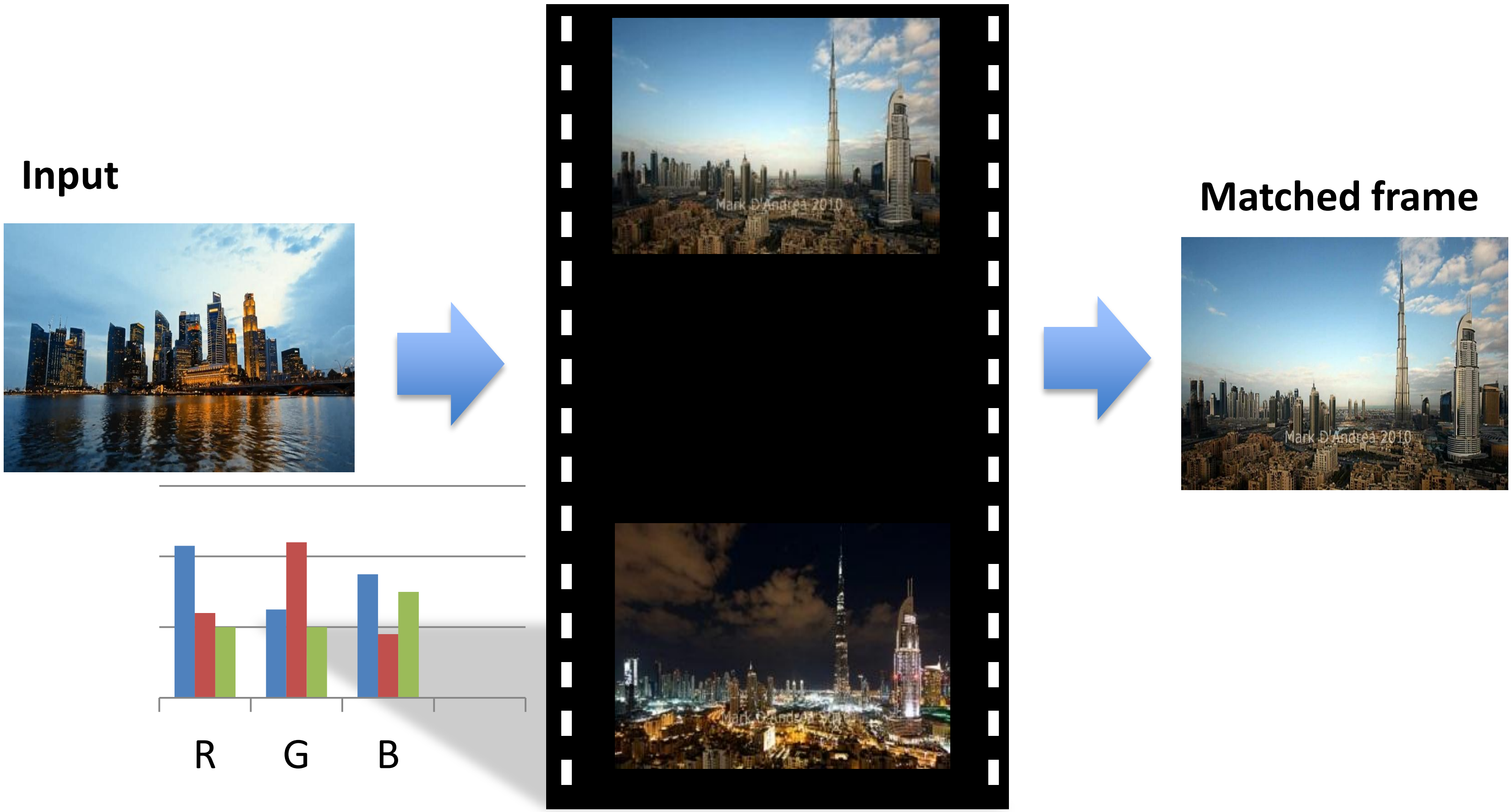
Matching step 1: video level

- Video retrieval with off-the shelf scene matching technique [Xiao et al, 2010]



Matching step 2: frame level

- Select the best match frame by color histogram metric

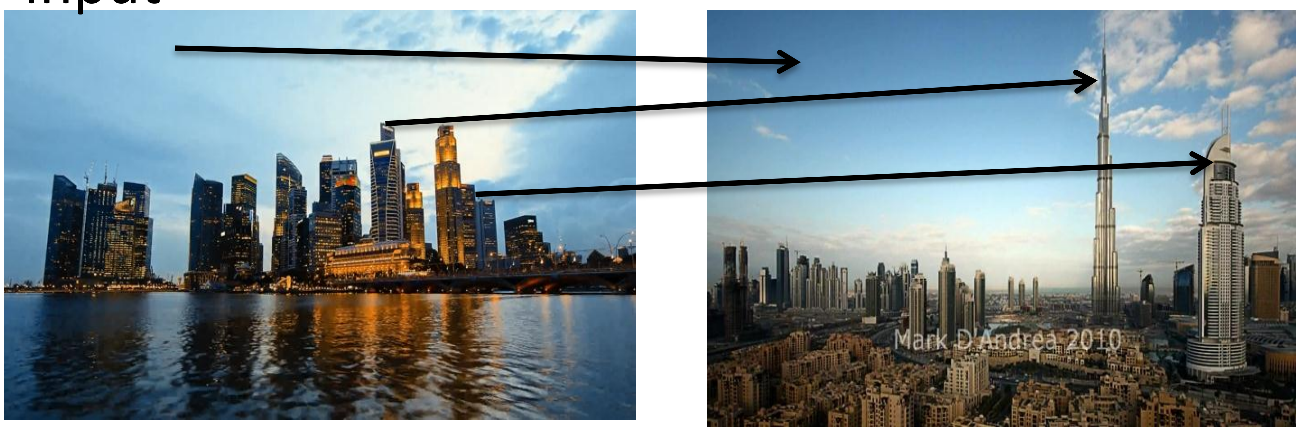


Matching step 3: pixel level

- Goal: respect scene semantic
  - E.g., sky to sky, building to building

Input

Matched frame



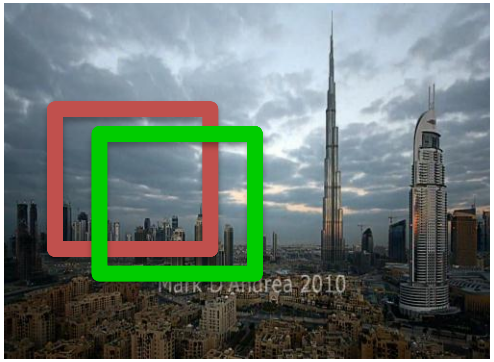
- Dense correspondence using Markov random field

Markov Random Field for dense matching

- Data term: standard L2 norm
- Regularization term: aggregate over the entire sequence, not just the matched frame
  - Consistency over all time of day

max (

,



…

)

Validating the matching

- Warp the matched frame to input using the dense correspondence

warp

Naïve transfer:
warp the target frame to the input

- Using the same correspondence

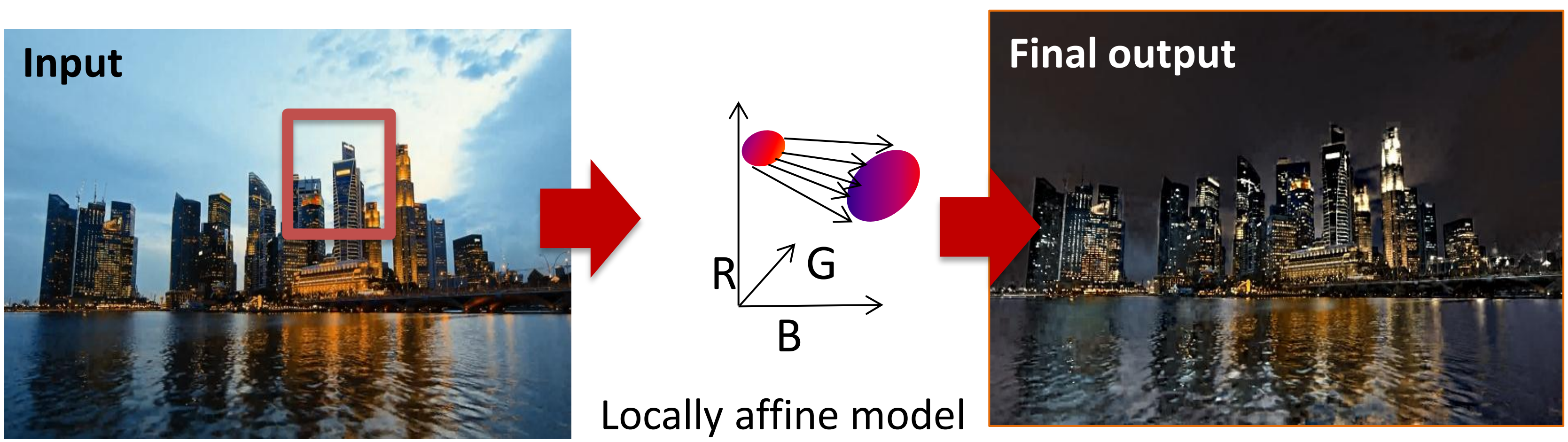
- The texture in the warped image is wrong

- Actually, the input already told us the texture.

Our approach: locally affine color transfer

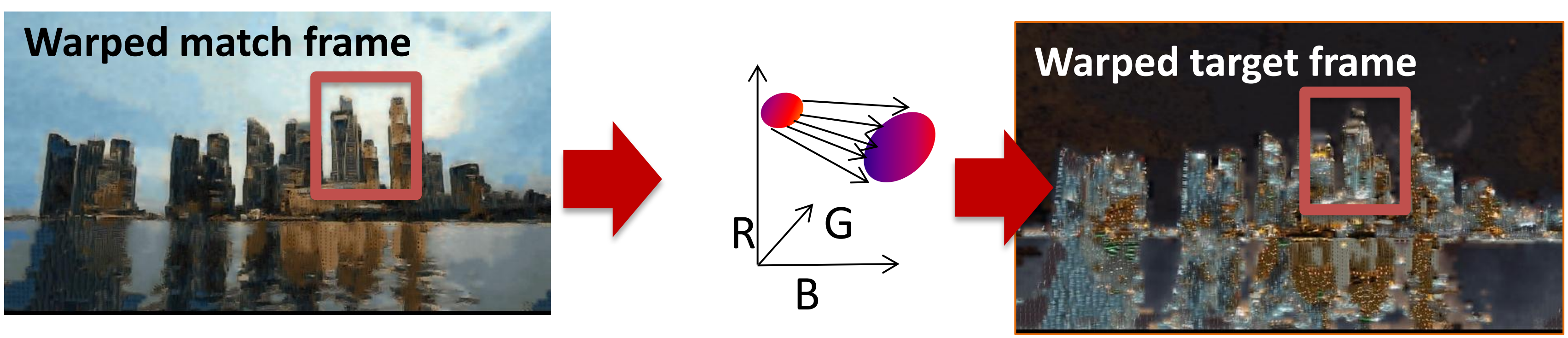
- *Local* to handle complex scenes

- *Affine* color transfer in each patch
  - preserve the structure of the input
  - match ground truth data; see paper.



Locally affine model explains the color change of time-lapse data

- In particular, explain matched frame and target frame



Locally affine model

The transfer needs to be locally affine everywhere

- The patches are overlapping, so we cannot estimate the affine model independently on each patch

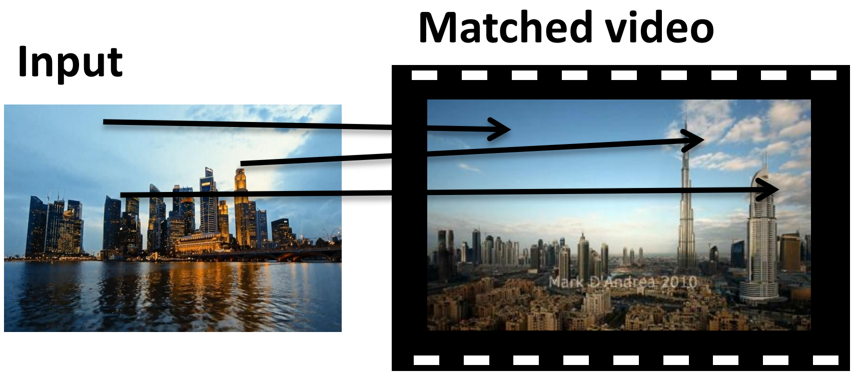
Color transfer as an optimization

- We are looking for color remapping function
  - Objective #1: explain time-lapse data
  - Objective #2: locally affine everywhere

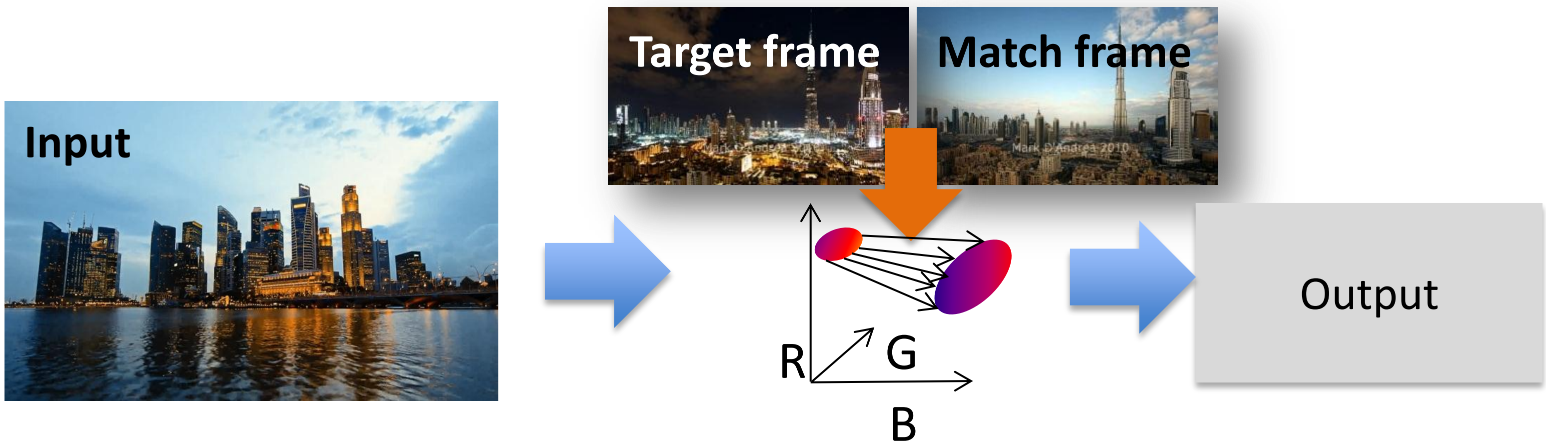
- We design a least-squares energy
  - Sparse linear system
  - Formula and detailed analysis in the paper

Recap

1. Match input to video
   i. scene matching
   ii. frame matching
   iii. dense matching



2. Locally affine color transfer



Locally affine model

Input at sunset

Input at sunset

Our result at night

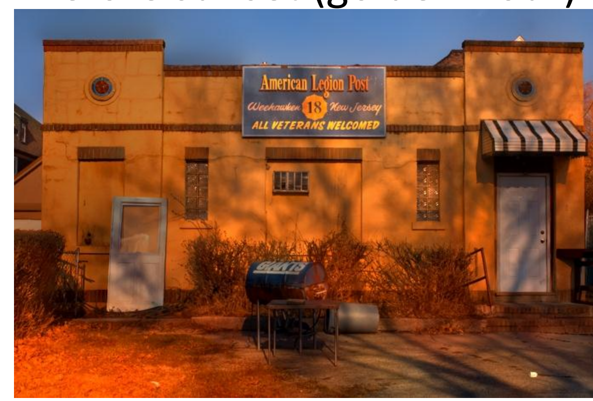
Results

- Same input for four different times of day

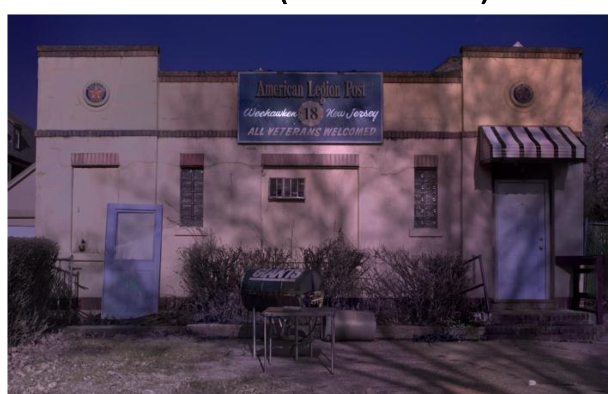
Input

Day

Before sunset (golden hour)



After sunset (blue hour)



Night

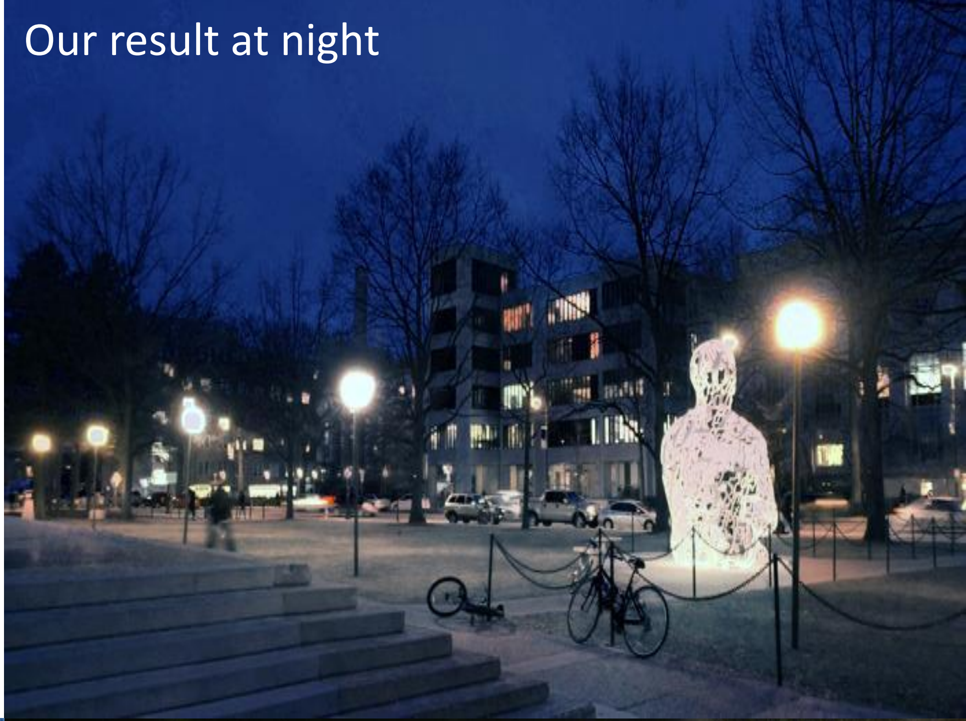

Day
American Legion Post
Weehawken 18 New Jersey
ALL VETERANS WELCOMED

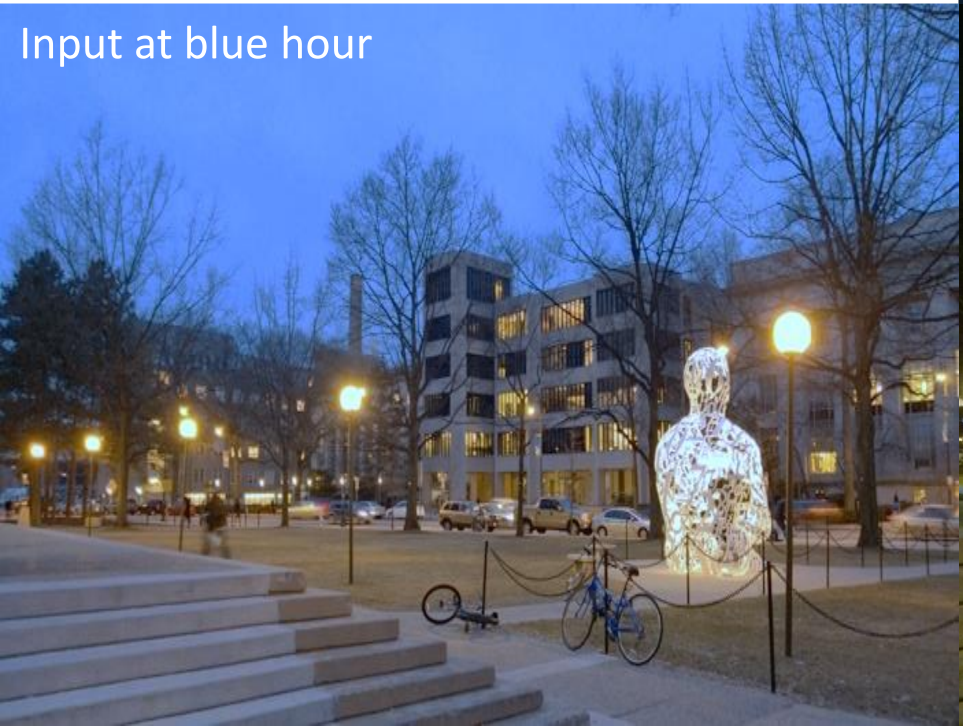

Before sunset (golden hour)
American Legion Post
Weehawken 18 New Jersey
ALL VETERANS WELCOMED


After sunset (blue hour)
American Legion Post
Weehawken 18 New Jersey
ALL VETERANS WELCOMED


Night
American Legion Post
Weehawken 18 New Jersey
ALL VETERANS WELCOMED

Ground truth validation

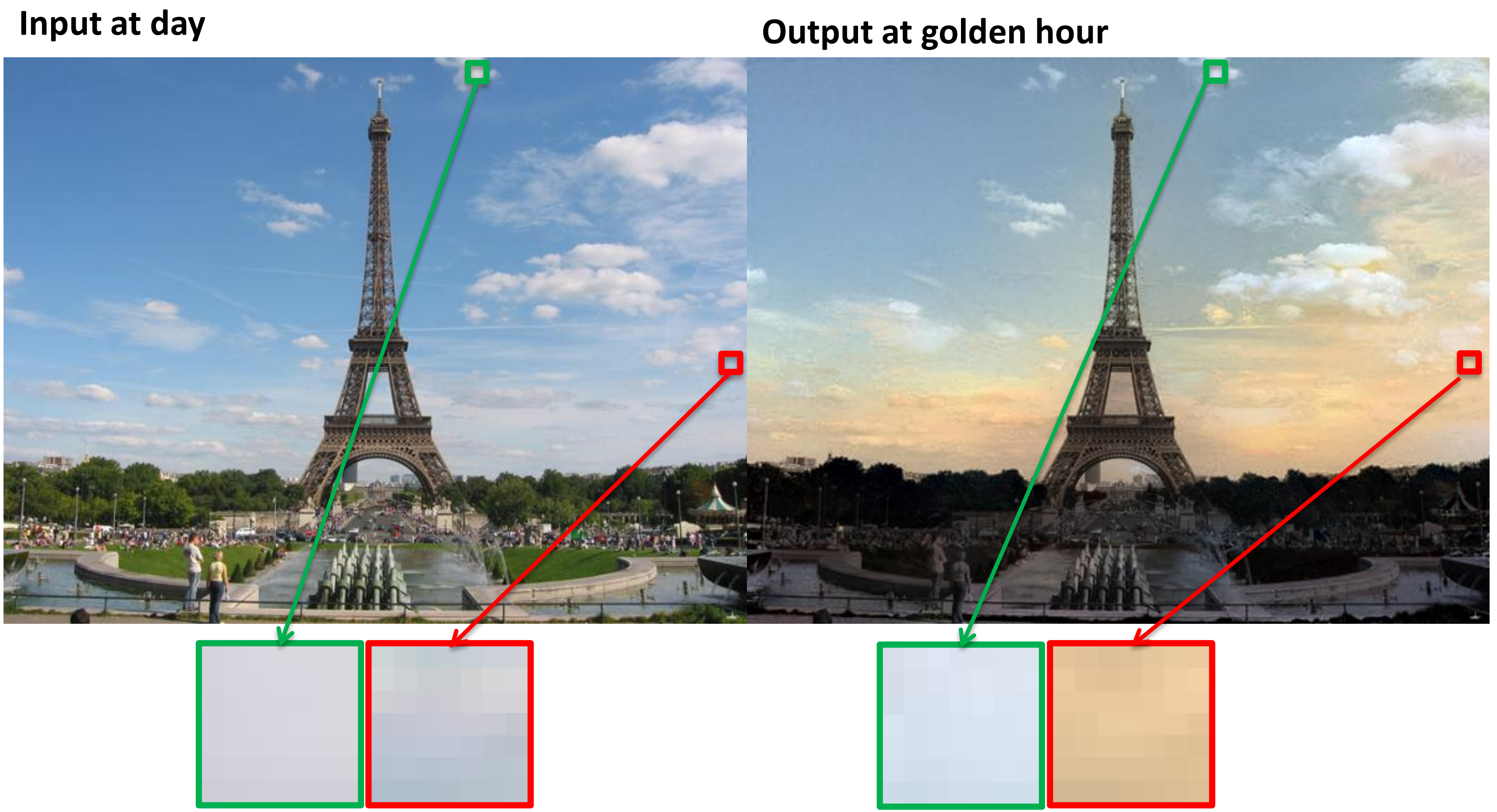

Our result at night


Input at blue hour


Ground truth at night

Our transfer is spatially-variant

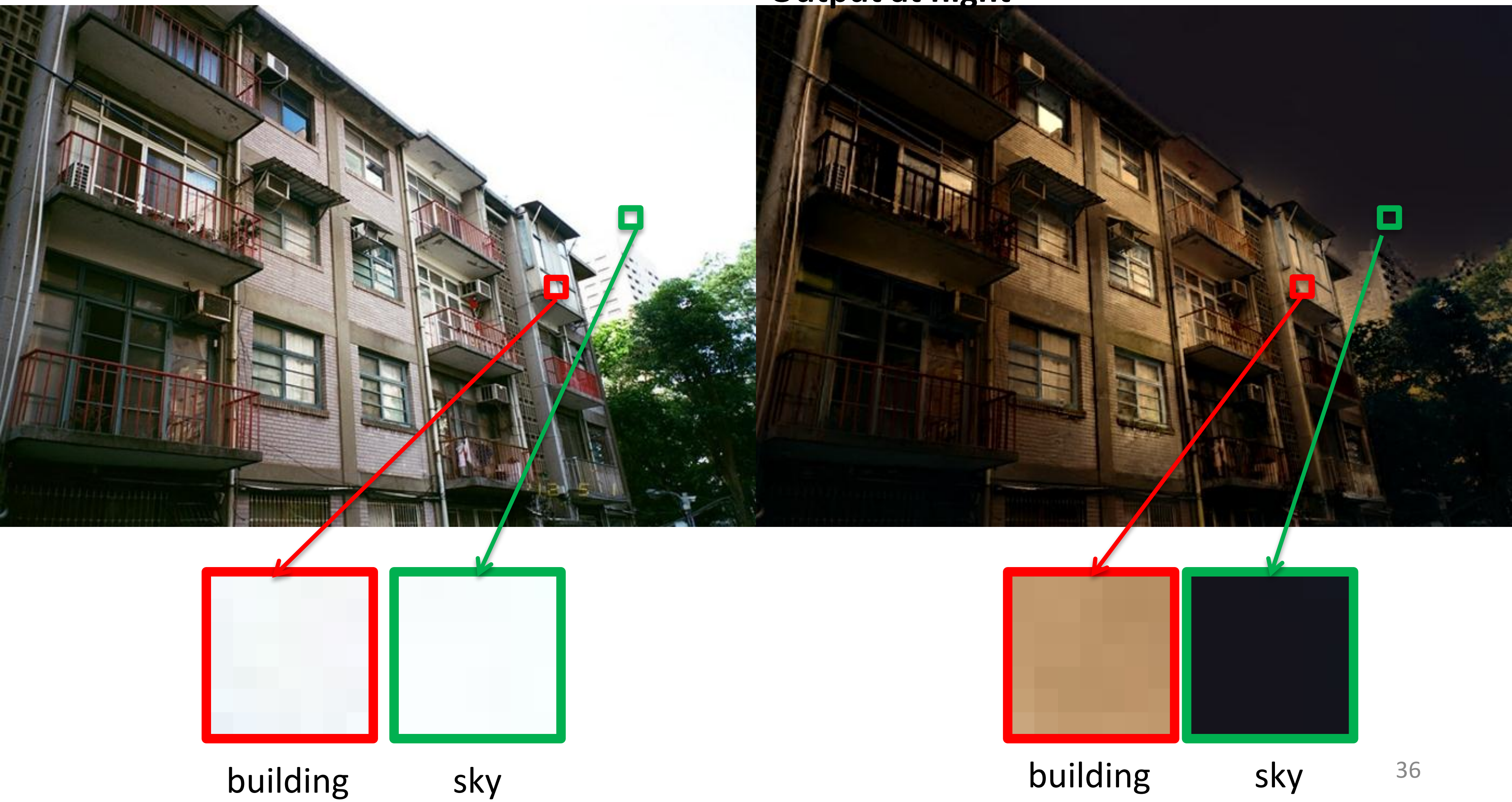
- Our transfer is local.



Our transfer is object-dependent

- We respect semantic in the scene.

Input at day **Output at night**



Run-time Performance

- Image size: 700-pixels width.
- Matching takes 25 seconds
  - 2 seconds for scene matching.
  - 23 seconds for dense correspondence
- Locally affine transfer takes 32 seconds.
- Implemented with unoptimized Matlab

Various input image types: cloudy

Cloudy input

Output at after sunset

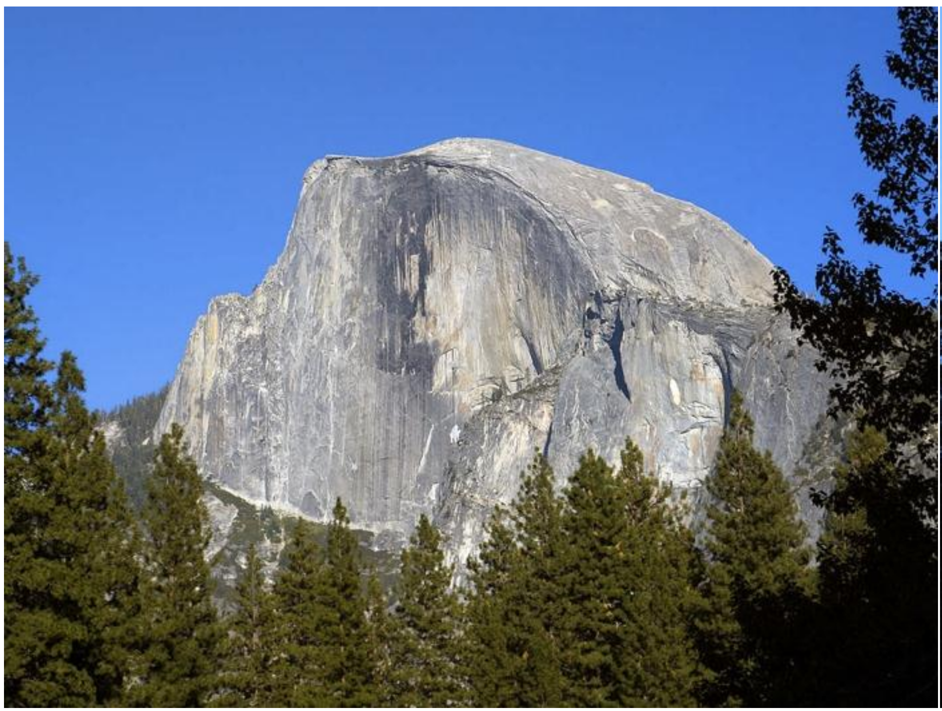
Various input image types: after sunset

Input at after sunset

Output at night

Results for different input scenes: mountain

Input at day



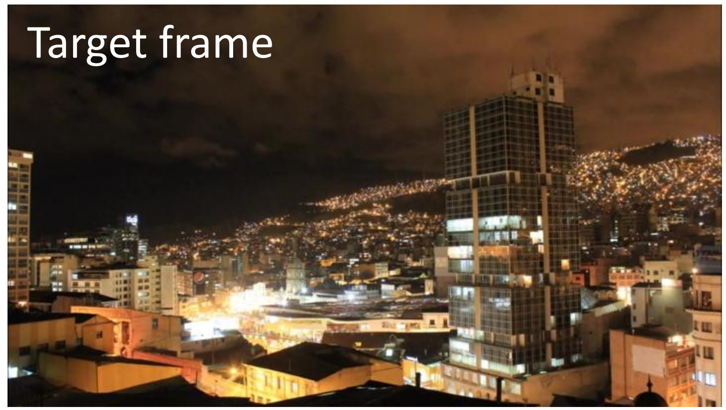
Output at blue hour

Results for different input scenes: lake

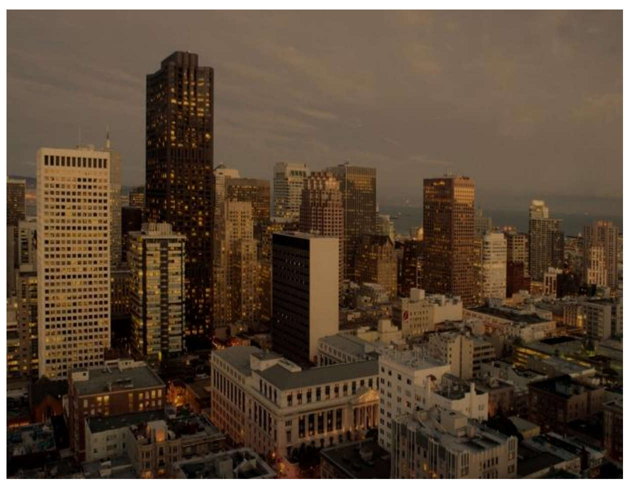
Input at day

Output at night

Comparison with other methods: translate to night



[Pitié et al. 2005]



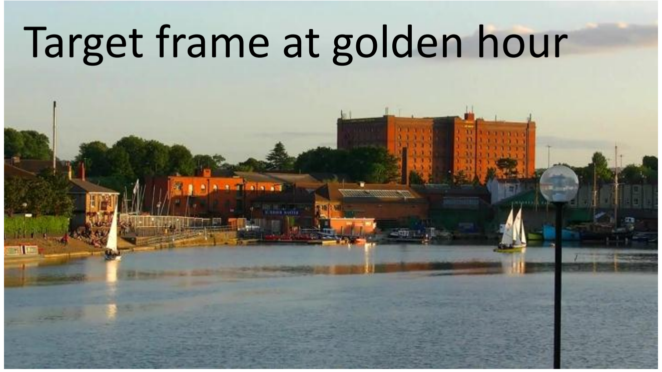
[Reinhard et al. 2001]

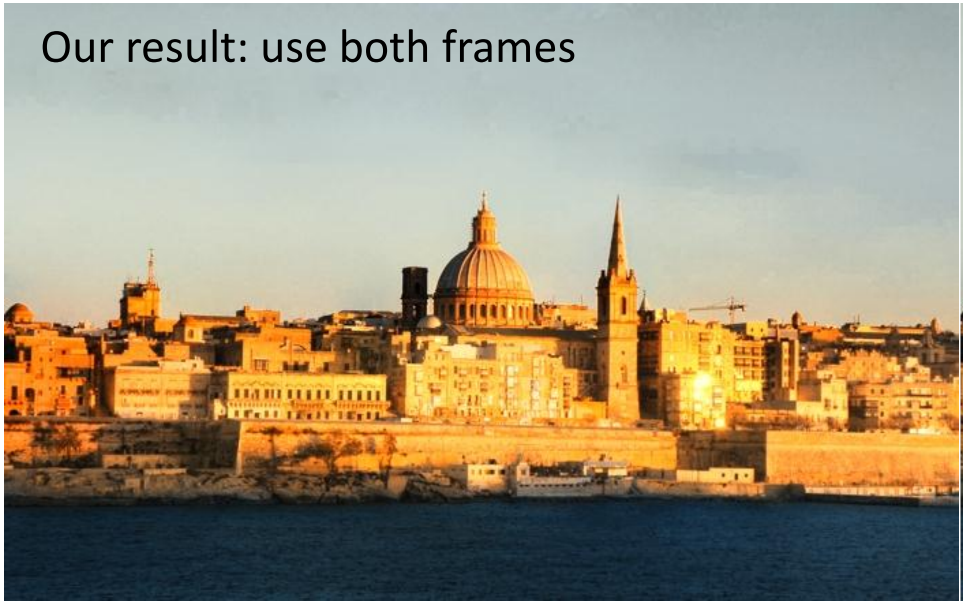
Our method


Input
Our method
[Reinhard et al. 2001]
[Pitié et al. 2001]

Color Transform vs Color Distribution

- Our result is more golden





Application: continuous control

Containing videos

Application: translate the time of day of a painting

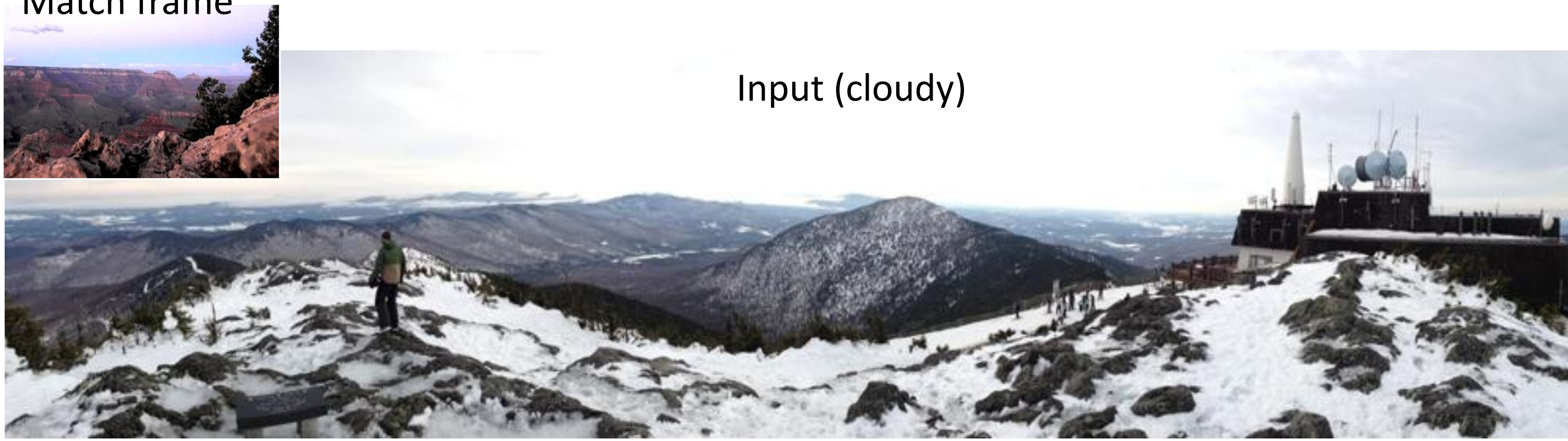
Input at day

Output at blue hour

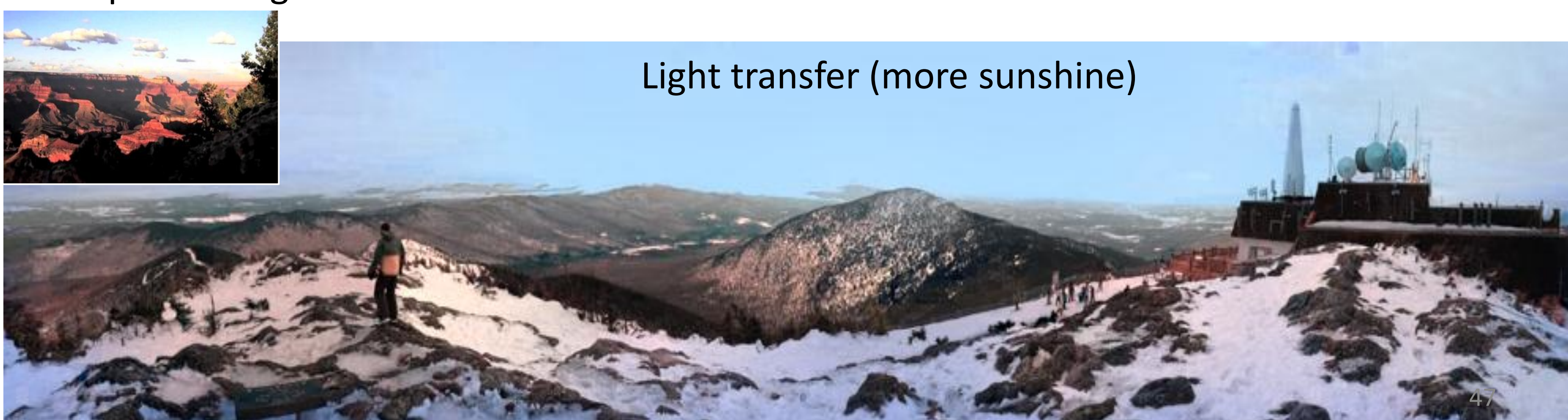
“In the Auvergne”, Jean-Francois Millet

Application: lighting transfer

Match frame



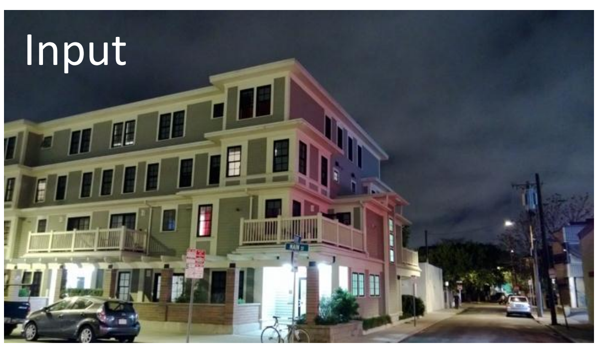
Hand-picked target frame

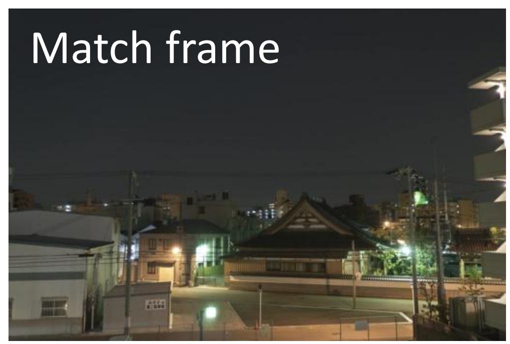


Limitations

- Dynamic scenes are challenging
- We do not turn on lights
- Night-to-day case does not work well
- Plausible but not physically accurate

Night to Day





Output: color and shadow are wrong

Conclusion

- We introduce time hallucination: render an image at another time of day
- We use a time-lapse database, and propose a locally affine model to transfer the color change between two frames

Acknowledgments

- We thank Jianxiong Xiao for the help and advice in scene matching code, SIGGRAPH ASIA reviewers for their comments, and acknowledge the funding from NSF No.0964004 and NSF CGV-1111415.