

# Algorithm to distinguish reflections from photos taken through windows

May 11 2015, by Larry Hardesty



MIT researchers have created a new algorithm that, in a broad range of cases, can automatically remove reflections from digital photos. On the left is the original photo taken through a window, with the photographer's reflection clearly visible. On the right, the reflection has been separated from the photo.

It's hard to take a photo through a window without picking up reflections

of the objects behind you. To solve that problem, professional photographers sometimes wrap their camera lenses in dark cloths affixed to windows by tape or suction cups. But that's not a terribly attractive option for a traveler using a point-and-shoot camera to capture the view from a hotel room or a seat in a train.

At the Computer Vision and Pattern Recognition conference in June, MIT researchers will present a new algorithm that, in a broad range of cases, can automatically remove reflections from digital photos. The algorithm exploits the fact that photos taken through windows often feature two nearly identical reflections, slightly offset from each other.

"In Boston, the windows are usually double-paned windows for heat isolation during the winter," says YiChang Shih, who completed his PhD in computer science at MIT this spring and is first author on the paper.

"With that kind of window, there's one [reflection](#) coming from the inner pane and another reflection from the outer pane. But thick windows are usually enough to produce a double reflection, too. The inner side will give a reflection, and the outer side will give a reflection as well."

Without the extra information provided by the duplicate reflection, the problem of reflection removal is virtually insoluble, Shih explains. "You have an image from outdoor and another image from indoor, and what you capture is the sum of these two pictures," he says. "If  $A+B$  is equal to  $C$ , then how will you recover  $A$  and  $B$  from a single  $C$ ? That's mathematically challenging. We just don't have enough constraints to reach a conclusion."

## Thinning the field

The second reflection imposes the required constraint. Now the problem becomes recovering  $A$ ,  $B$ , and  $C$  from a single  $D$ . But the value of  $A$  for one pixel has to be the same as the value of  $B$  for a pixel a fixed distance

away in a prescribed direction. That constraint drastically reduces the range of solutions that the algorithm has to consider.

Nonetheless, a host of solutions still remain. To home in on one of them, Shih and his coauthors—professors of [computer science](#) and engineering Frédo Durand and Bill Freeman, who were Shih's thesis advisors, and Dilip Krishnan, a former postdoc in Freeman's group who's now at Google Research—assume that both the reflected image and the image captured through the window have the statistical regularities of so-called natural images.

The basic intuition is that at the level of small clusters of pixels, in natural images—unaltered representations of natural and built environments—abrupt changes of color are rare. And when they do occur, they occur along clear boundaries. So if a small block of pixels happens to contain part of the edge between a blue object and a red object, everything on one side of the edge will be bluish, and everything on the other side will be reddish.

In [computer vision](#), the standard way to try to capture this intuition is with the notion of image gradients, which characterize each block of pixels according to the chief direction of color change and the rate of change. But Shih and his colleagues found that this approach didn't work very well.

## Playing the odds

Ultimately, they settled on a new technique co-developed by Daniel Zoran, a postdoc in Freeman's group. Zoran and Yair Weiss of the Hebrew University of Jerusalem created an algorithm that divides images into 8-by-8 blocks of pixels; for each block, it calculates the correlation between each pixel and each of the others. The aggregate statistics for all the 8-by-8 blocks in 50,000 training images proved a

reliable way to distinguish reflections from images shot through glass.

In their paper, Shih and his colleagues report performing searches on Google and the Flickr photo database using terms like "window reflection photography problems." After excluding results that weren't photos shot through glass, they had 197 images, 96 of which exhibited double reflections that were offset far enough for their algorithm to work.

"People have worked on methods for eliminating these reflections from photos, but there had been drawbacks in past approaches," says Yoav Schechner, a professor of electrical engineering at Israel's Technion. "Some methods attempt using a single shot. This is very hard, so prior results had partial success, and there was no automated way of telling if the recovered scene is the one reflected by the window or the one behind the window. This work does a good job on several fronts."

"The ideas here can progress into routine photography, if the algorithm is further robustified and becomes part of toolboxes used in digital photography," he adds. "It may help robot vision in the presence of confusing glass reflection."

**More information:** "Reflection Removal Using Ghosting Cues."  
[dilipkay.wordpress.com/reflection\\_ghosting/](http://dilipkay.wordpress.com/reflection_ghosting/)

*This story is republished courtesy of MIT News  
([web.mit.edu/newsoffice/](http://web.mit.edu/newsoffice/)), a popular site that covers news about MIT  
research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: Algorithm to distinguish reflections from photos taken through windows (2015, May

11) retrieved 20 March 2024 from <https://phys.org/news/2015-05-algorithm-distinguish-photos-windows.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.