Super-resolution with Epitomes

Aaron Brown

University of Wisconsin

Madison, WI

Abstract

Techniques exist for aligning and stitching photos of a scene and for interpolating image data to generate higher-resolution images. We show how to use image epitomes to increase the resolution of images with the aid of high resolution sample data. Applications include generating stills from movie clips and using low-quality image sensors, ubiquitous to mobile devices, to capture photos.

Introduction

This paper examines how we can use statistical techniques to improve the quality of an image, namely to increase the resolution of a given image.

Motivation

Current techniques for upscaling images and interpolating finer details do not give great results. It is very difficult to reconstruct details of a scene from a low resolution image because the necessary information simply is not available. However, if we are given some hint or example of what the details of the scene should look like, we should be able to better reconstruct a high-resolution version of the given image.

This technique leads to several possible applications:

- **Video sequences**: video sequences are often recorded at much lower resolution than still images, however if the sequence includes zooming, some of the frames will be closeups of the details of the scene. Combining the detail frames of with a wide-angle shot, we should be able to construct a high-resolution version of the wide-angle shot.
- **Camera phones**: many hand-held devices, such as camera-phones, are very handy for taking photos, but have poor quality image sensors. However, these devices are often easy to program. If we can develop a system to automatically capture both the desired photo as well as necessary detail images, we can enhance photo quality without using more expensive sensors or lenses.
- **Compression:** Before compression, an image could be reduced to a low-resolution copy of itself, as well as some amount of detail information. The low-resolution image and the detail information could be compressed using existing image compression techniques, resulting in an overall reduction in image file size.
- **Sample library**: If we can appropriately categorize an image, we could use existing libraries of high resolution samples of images which are of scenes similar to the given image. The high resolution samples could then be used to enhance the quality of the given image. This could be especially effective for highly constrained images, such portraits or images of text.

Problem

Given a low-resolution image and a high resolution sample or set of samples of detail of the scene, output a high resolution reconstruction of the original low-resolution image that matches the original scene as closely as possible.

It may be ill posed to try and match the output to the original scene. For some applications, it may be enough for the output to look realistic or believable.

Related work

Some of the concepts used are similar to image quilting, as in Efros and Freeman [1]. Cheung et.al. [2] provides a good description of epitomes and how they can be used for video applications. Matlab code for this paper was based on source code provided by Cheung [3].

Theory

An epitome is a representation of the statically significant features of a dataset, such as an image, sound clip, or video. In the case of images, the epitome of an image will itself be an image, but usually much smaller in size than the original.

Epitomes are patch based, and once learned they can be used to fill in unknown pixels of an image in a manner similar to image quilting [1]. However, using epitomes is more robust in situations where the known pixels may vary from the training data. Epitome-based reconstruction also allows us to specify a confidence level for known pixels rather than just a binary known or unknown.

In order to create the epitome, the learning algorithm is given the training data. We then iterate over possible patch selections from the training sample and attempt to use the epitome to reconstruct the sample. We continue iterating and attempt to minimize the difference between the original and reconstructed sample. After several iterations, we should reach a local optima which satisfactorily captures the essence of the original sample.

Once the epitome is learned, we can use it to reconstruct an image. To do this we start provide the image data as well as confidence levels for each pixel in the image. The epitome reconstruction algorithm will apply combinations of patches to each area in the image where the patch data matches the image data that has been marked as high confidence.

By varying how the confidence level or variance is set, we can achieve different results. Cheung et. al. [2] showed how to use epitomes for tasks such as denoising, inpainting, dropped frame recovery, and super-resolution. For super-resolution, we scale up the input image by some value *n*, simply copying the image data to fill in missing pixels. We then every *n*th row and every *n*th column as high confidence, and the other pixels as low confidence.

Method

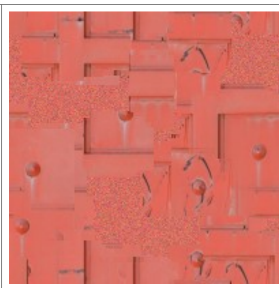
The epitome learning and reconstruction algorithms were implemented in Matlab and run on a 2.66GHz Intel Core 2. Implementation was based largely on Cheung et. al. [3] code for denoising an image.

To perform a test, a single high-resolution image was used. A small portion of the image was used as the high resolution sample. Both the sample and full image were scaled to 128x128 pixels for input to the algorithms. The low-resolution 128x128 image was then scaled up by the reconstruction algorithm to

512x512 pixels. Epitome size was set to be 100x100 pixels and patch size was initialized to 10x10 pixels. Processing time was about 10-15 minutes for each image.

Experimental results

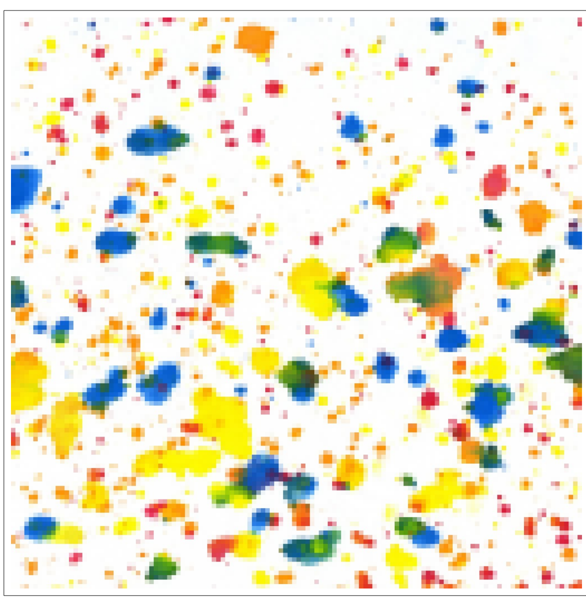
The first test was of some red doors. Results are very good, however there is a kink in the metal that is not in the sample, so it is replaced in the output. If we include that region of the door in the sample as well, we get better results.

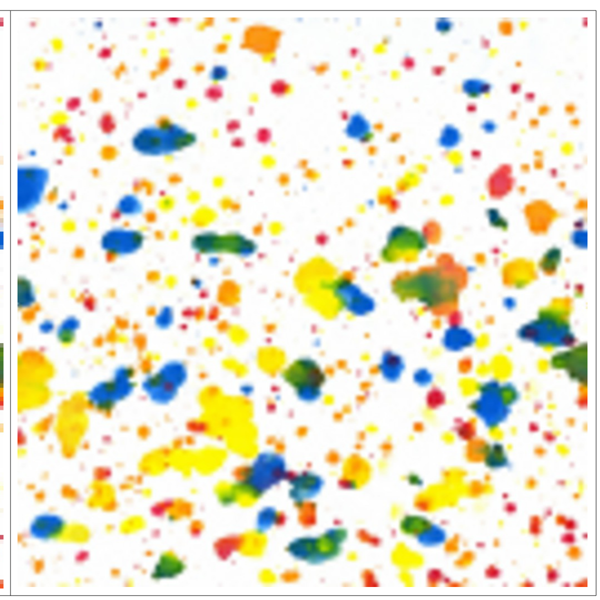
	
High-res sample	Epitome

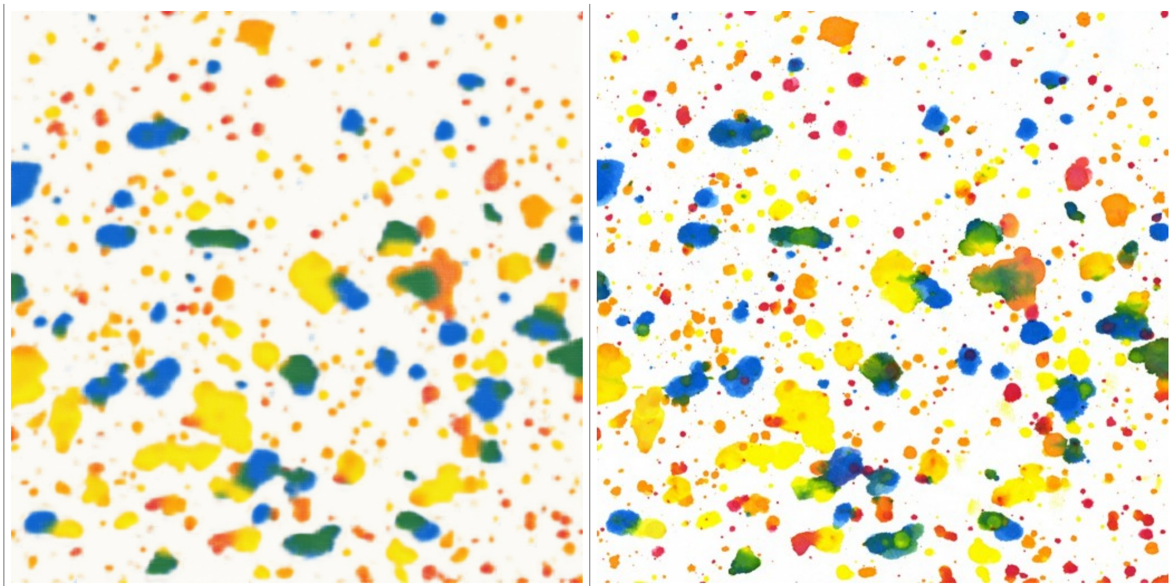
	
Low resolution input	Output

output using additional sample data

A second example of a paint splatter also generates good results. Note how well the algorithm copes with the irregular shapes and arrangement of the paint blotches. This example works well because we can capture the essence of the details of the image by looking at even just a small portion of the overall image.







A second experiment with an image of a paint splatter.
Clockwise, from top left: low resolution image, scaled image using “cubic” interpolation in GIMP, original high resolution image, reconstructed image

Concluding remarks

Super-resolution using epitomes works well for images which are uniform and regular. However, there are some disadvantages. Most images do not fit this description and we would not be able to easily select a region to use for high resolution sample. Current implementations are too slow for practical use. Features which are not captured by the sample will be replaced in the output. In general, this process produces output which may not be accurate, but is hopefully believable based on the sample.

Some potential improvements could be made to this process:

- Combine with other interpolation techniques: create some method to estimate the accuracy of areas in the result, and then blend the output with some other interpolation technique.
- Automatically capture sample data: by adding additional sample data to the doors image, we were able to improve the quality of the output. This

process could be generalized to automatically identify and capture important regions for sampling.

- Automatic registration: automatically determine the scale and orientation of sample data in relation to the input image, so that the process works in more general situations.

References

[1] A. Efros, and W. Freeman. Image Quilting for Texture Synthesis and Transfer. Proceedings of SIGGRAPH '01, Los Angeles, California, August, 2001.

[2] V. Cheung, B. J. Frey, and N. Jojic. Video epitomes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[3] Vincent Cheung. Image Epitome Example (Matlab source code). September, 2005.