Super-Resolution Imaging

Stéfan van der Walt

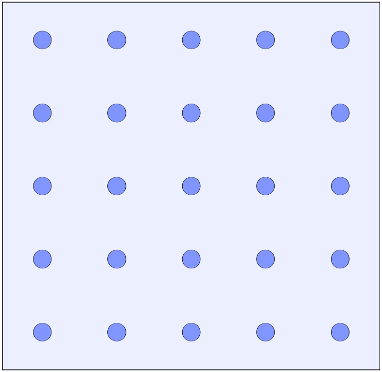
Computer Vision & Machine Learning at Stellenbosch (VLS)

26 February 2010

Launch Day Blues

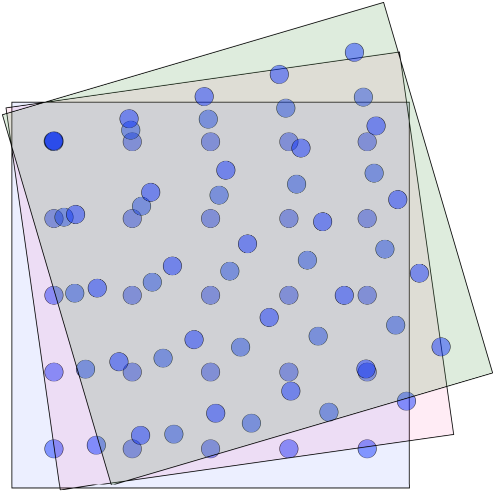
- We're talking about image super-resolution (not radar).
- This is not CSI!
- Launch Day: did we remember everything?
- Camera systems used at their limits
- We always want to do better – with a little help we can

Data is Crucial



Say our image resolution is not high enough.

- If we're stuck with one picture we're stuck
  - Single-frame "super-resolution"
  - Even with 100 pictures of the same scene we're not much better off
- Can we improve the sampling rate?
  - Can't measure a 7Hz signal with a 5Hz sampler, but what if we had 3?



With many images from slightly different viewpoints/angles, there's hope!

Image Formation Headaches

During image formation a number of unpleasant effects rear their heads:

- Lens distortions
- Sensor noise
- Parallax
- Demosaicking (Bayer pattern) We own one of the few Foveon X3 sensors in South Africa!

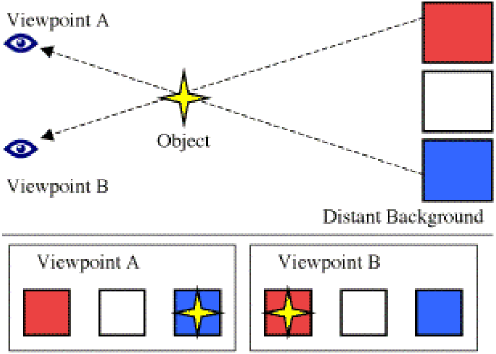


Image from http://commons.wikimedia.org/wiki/File:Parallax_Example.png

Problem Solving 101

- The easiest way to get rid of a problem is to define it away.
- In the applied world, this is difficult: data doesn't lie.
- Next best option: make assumptions (pick our data carefully).

Assumptions

- Perform SR on small area (no lens distortion)
- Large distance to object (no parallax, frames related by a homography)
- No diffraction limiting
- Probably many other implicit assumptions...

The Super-Resolution Model

We are dealing with, say, 30 odd frames. The image acquisition process for a single frame, i, is often represented as the simplified model

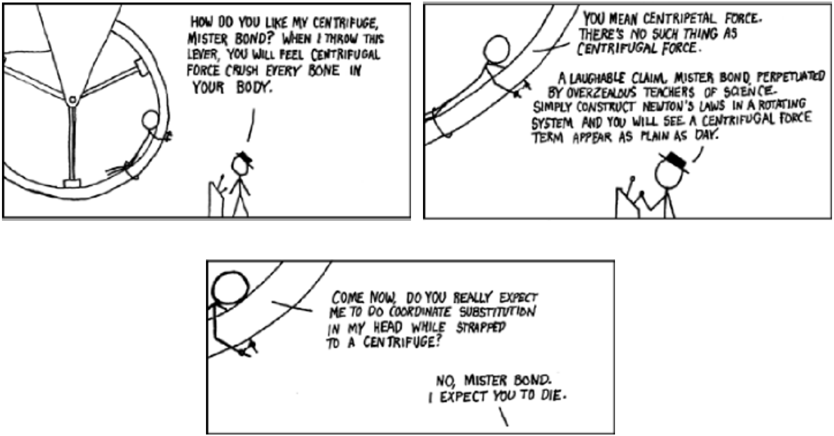
$$\mathbf{b}_i = S\downarrow(h(\mathscr{T}_i(\mathbf{x}))) + \eta_i$$

where

- $\mathbf{b}_i$ is the i-th low-resolution (LR) frame,
- $\mathbf{x}$ is a high-resolution (HR) representation of the scene,
- $\mathscr{T}_i$ is a geometric transformation for frame i dependent on camera position,
- h is the camera point-spread function,
- $S\downarrow$ is the downsampling operator and
- η_i is additive normally distributed noise for frame i.

Expectations

What do you expect to achieve with such a generic model?



From http://xkcd.com/123/

Expectations

We simplify drastically by linearising from

$$\mathbf{b} = S\downarrow(h(\mathscr{T}(\mathsf{x}))) + \eta$$

to

$$\mathbf{b} = A\mathsf{x} + \eta.$$

Importantly, A represents downsampling and geometric distortion as well as PSF.

Are we now in a better position to find a solution?

The camera matrix, A

For each frame i, we have

$$\mathbf{b}_i = A_i \mathbf{x} + \eta.$$

We can neglect the assumed zero-mean Gaussian noise and combine all these linear equations to obtain a least squares problem

$$A\mathbf{x} = \mathbf{b}$$

where

$$A = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{n-1} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{n-1} \end{pmatrix}.$$

Detour: Bayesian approach

Model the problem as

$$P(\mathbf{x}|\mathbf{b},\mathbf{c})$$

where $\mathbf{c}$ represents all camera parameters. Use Bayes's rule to rewrite as

$$P(\mathbf{x}|\mathbf{b},\mathbf{c}) = \frac{P(\mathbf{b}|\mathbf{x},\mathbf{c})P(\mathbf{x}|\mathbf{c})}{P(\mathbf{b}|\mathbf{c})}.$$

Modelling pixels as normally distributed along true values, assuming linear relationship between $\mathbf{x}$ and $\mathbf{b}$, we derive the solution

$$\arg\min_{\mathbf{x}} \left[\|\mathbf{b} - A\mathbf{x}\|^2 + \lambda \mathbf{x}^T \mathbf{x} \right] \qquad \text{(compare to } A\mathbf{x} = \mathbf{b}\text{).}$$

This corresponds to the damped least squares solution (the damping is due to the prior $P(\mathbf{x}|\mathbf{c})$).

Structure of matrix A

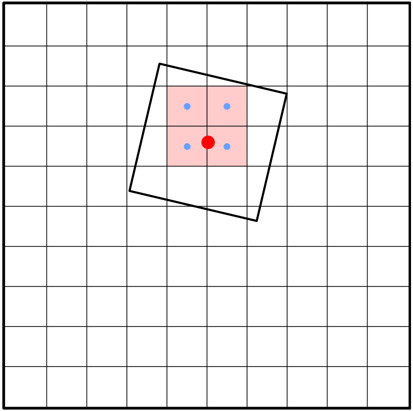
In $A\mathbf{x} = \mathbf{b}$, each row of A contains weights for the pixels of $\mathbf{x}$ that reproduce a single pixel in $\mathbf{b}$.
Recall that A represents

$$S \downarrow (h(\mathscr{T}(\mathbf{x}))).$$

Simplify A further to only represent geometric transformation and interpolation. Since we neglect the camera PSF, so we expect to run into problems (and we soon do—sketch forshortening).

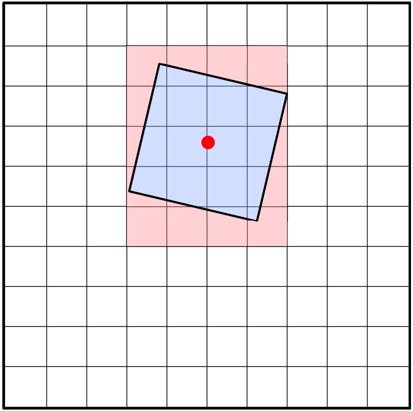
- For today, we assume that the geometric transformations between frames are known. In reality, this has to be estimated via image registration (alignment). We have developed some interesting discrete pulse transform-based feature detectors to assist with this task.

Camera matrix: bilinear interpolation



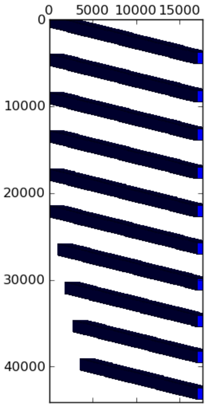
Footprint is not large enough—certain high-resolution pixels now totally unrelated to low-resolution pixels.

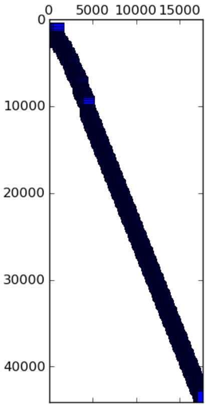
Camera matrix: polygon interpolation



Footprint covers all important neighbours—relationship established between high and low-resolution pixels. Still a linear operator!

Non-zeros in A (polygon interpolation)





$$\begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{n-1} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{n-1} \end{pmatrix}$$

Solving

We're left with the minimisation problem

$$\arg\min_{\mathbf{x}} \left[\|\mathbf{b} - A\mathbf{x}\|^2 + \lambda \mathbf{x}^T \mathbf{x} \right].$$

- Direct or iterative? Ill-posed problem; we really need the damping. Use iterative method.

1. Steepest descent
2. Conjugate gradient (we know the gradient of the 2-norm-squared)
3. LSQR (makes use only of products $A\mathbf{x}$ and $A^T\mathbf{x}$).

A Better Prior

The second term in

$$\arg\min_{\mathbf{x}} \left[\|\mathbf{b} - A\mathbf{x}\|^2 + \lambda \mathbf{x}^T \mathbf{x} \right]$$

keeps $\mathbf{x}$ close to zero. We have a better guess for $\mathbf{x}$ than zero, namely the stack (average) of all aligned, upscaled low-resolution images: $\mathbf{y}$ (this is a cheap "super-resolution" estimate). We can rewrite the problem in terms of the error as

$$\hat{\mathbf{b}} = \mathbf{b} - A\mathbf{y}.$$

If $\mathbf{y}$ is a good estimate of $\mathbf{x}$, then the solution to

$$\arg\min_{\hat{\mathbf{x}}} \left[\left\| \hat{\mathbf{b}} - A\hat{\mathbf{x}} \right\|^2 + \lambda \hat{\mathbf{x}}^T \mathbf{x} \right]$$

lies around zero. We find our final solution as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{y}.$$

The 2-Norm in High Dimensions

The behaviour of norms change as dimensionality increases.

Volume of sphere in D dimensions:

$$V(r) = \frac{\pi^{\frac{D}{2}} r^D}{\Gamma\left(1+\frac{D}{2}\right)}$$

Volume in outer shell of thickness ε:

$$\frac{V(r) - V(r(1-\varepsilon))}{V(r)} = \frac{1^D - (1-\varepsilon)^D}{1^D}.$$

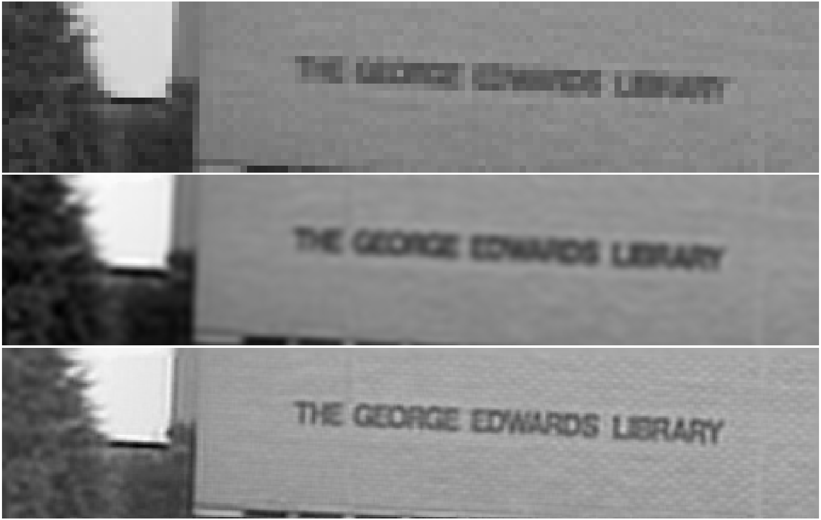
This becomes 1 as $D \to \infty$.

Weakness of norms in high-dimensions: All error vectors $\mathbf{v}_i$, relative to an image $\mathbf{f}$, of the form $\mathbf{v}_i = (\mathbf{f} - \mathbf{g}_i)$ lie close to the outer surface of the hypersphere centred around $\mathbf{f}$ – their p-norms, $\|\mathbf{v}\|_p$, are therefore all very similar.

Demo / results

Input, 1.8x bilin, 5x bilin, 5x polygon

Demo / results



Input (upscaled) 1/30, stack, poly

Software

- Results from a highly-complex software system cannot be trusted unless you've inspected the software.
- Free software library containing SUper REsolution MEthods at `http://mentat.za.net/supreme/`

Image registration (mutual information, sparse) • Warping (affine, log-polar, etc.) • Feature detectors (DPT, FAST and KLT) • Discrete Pulse Transform • Polygon operations • RANSAC (LO-RANSAC, MSAC) model fitting • Wavelet denoising • Fast Summed Area Table template matching • Chirp-Z Transform • Large least-squares solvers (Steepest Descent, CG, LSQR)