Super Resolution for Automated Target Recognition

Maksim Levental

***Abstract*—Super resolution is the process of producing high-resolution images from low-resolution images while preserving ground truth about the subject matter of the images and potentially inferring more such truth. Algorithms that successfully carry out such a process are broadly useful in all circumstances where high-resolution imagery is either difficult or impossible to obtain. In particular we look towards super resolving images collected using longwave infrared cameras since high resolution sensors for such cameras do not currently exist. We present an exposition of motivations and concepts of super resolution in general and current techniques, with a qualitative comparison of such techniques. Finally we suggest directions for future research.**

1 Image Registration

Images obtained from multiple vantage points, or at different times, of the same scene, become distorted with respect to each other. Since in MISR the aim is to exploit new information across multiple LR samples we need to first rectify these distortions and reconcile the images. Effectively this means finding one or more pixel transformations that enable mapping all LR images to a common pixel grid. When the transformations cannot be deduced from first principles (e.g. precise knowledge of the relative motion of the scene and the imaging system) they must be estimated from the LR images. There are broadly two perspectives on constructing the transformation f (see eqn. (**??**)): the global perspective which aims to model motion as a map of the image as a whole and the local perspective which aims to model motion as a deformation of individual pixels. These two perspectives naturally correspond to a globally or locally defined transformation f. We first cover global algorithms, examining the different techniques available for each step, and then move on to local algorithms.

1.1 Global Algorithms

Most global image registration algorithms consist of a feature detection and selection step (also called Control Point (CP) selection), a feature matching step, and a transform estimation step.

1.1.1 Feature Detection and Selection

Feature detection and selection is the process of identifying features of the image that are presumed to be invariant across the multiple images to registered. Note that here by features we mean image artifacts (e.g. edges, contours, line intersections, or corners); in this context encodings or transformations of these image artifacts are called *descriptors*. The CPs are the data that will be used to estimate the transformation f. Therefore, in order that the estimated transformation is accurate, CPs should be robust to noise and image degradation, sufficiently distributed throughout the image, and readily matched in the matching step.

1.1.1.1 Harris Corner Detection

Bentoutou *et al.***bentoutou2005automatic** use a Harris detector**harris1988combined** to find corner points, arguing that corners are robust to noise and stable over multiple images. The Harris detector improves on the Moravec**moravec1980obstacle** detector. The Moravec detector starts from the error function $E_{x,y}(u,v)$ which computes the sum of the squared differences (SSD) between an $m \times m$ weighted window around a pixel $X(x,y)$ and weighted windows shifted by u, v pixels:

$$E_{x,y}(u,v) \coloneqq \sum_{i,j=-m/2}^{m/2} w_{ij} \left[X(x_i+u, y_j+v) - X(x_i, y_j)\right]^2 \tag{1}$$

where $x_i \coloneqq x+i$ and $y_j \coloneqq y+j$. Moravec assigns a "corner score" according to the following reasoning (see figure 2):

1) If a pixel is in a region of uniform intensity then $E_{x,y}(u,v)$ is small for all u, v (since neighboring windows are similar).
2) If a pixel is on an edge, then $E_{x,y}(u,v)$ for either $u > 0$ or $v > 0$, but not both, is high.
3) If a pixel is on a corner, then $E_{x,y}(u,v)$ for $u > 0$ and $v > 0$ is high.

Therefore the corner score at pixel coordinate (x,y) is $\min_{u,v} E_{x,y}(u,v)$ in order to select for the third case. Moravec comments that this corner score is not isotropic, i.e. if edges aren't aligned with either the pixel axes or diagonals then $E_{x,y}(u,v)$ will incorrectly be low. Harris' insight was to linearize $E_{x,y}(u,v)$ in order to compute a quantity more closely related to the intensity variation in a local neighborhood of a pixel:

$$X(x_i+u, y_j+v) \approx X(x_i, y_j) + \frac{\partial X}{\partial u}u + \frac{\partial X}{\partial v}v \tag{2}$$

where the partial derivatives are taken at (x,y). This implies

$$E_{x,y}(u,v) \approx \sum_{i,j=-m/2}^{m/2} w_{ij} \left[X_u u + X_v v\right]^2 \tag{3}$$

$$= \sum_{i,j=-m/2}^{m/2} w_{ij} \left[X_u^2 u^2 + X_v^2 v^2 + 2X_u X_v uv\right] \tag{4}$$

$$= [u,v] \begin{bmatrix} \sum w_{ij} X_u^2 & \sum w_{ij} X_u X_v \\ \sum w_{ij} X_u X_v & \sum w_{ij} X_v^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{5}$$

$$= [u,v]\, M \begin{bmatrix} u \\ v \end{bmatrix} \tag{6}$$

where $X_u = \partial X/\partial u$ and similarly X_v. The matrix in eqn. (6), called the *structor tensor* or *second-moment matrix* M, is the

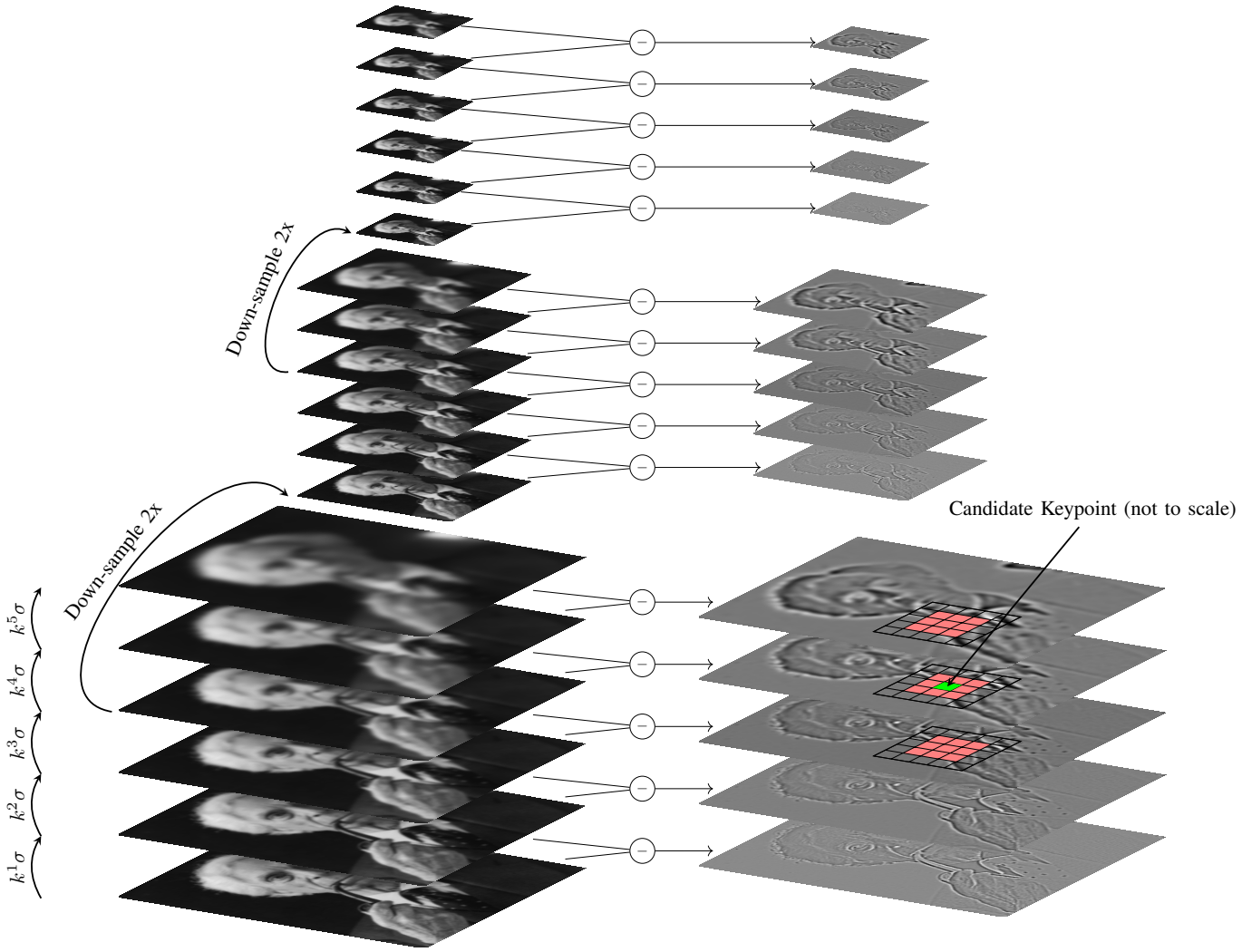


Fig. 1: The imaging model illustrating the relationship between a scene and final low-resolution images due to noise, motion, blur, and sampling.

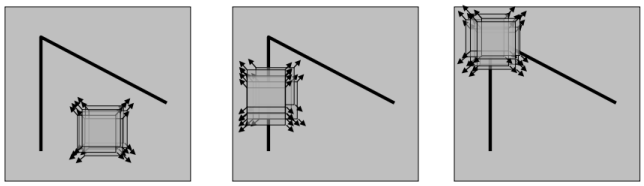


Fig. 2: Moravec Corner Detector

quantity Harris investigated. Harris reasoned that the cases of Moravec correspond to conditions on the eigenvalues λ_1, λ_2 of M:

1) If $\lambda_1 \approx \lambda_2 \approx 0$ then $X(x, y)$ is in a region of uniform intensity.
2) If $\lambda_1 \gg \lambda_2$ or $\lambda_2 \gg \lambda_1$ then $X(x, y)$ is on an edge.
3) $\lambda_1 \approx \lambda_2 > 0$ then $X(x, y)$ is on a corner.

Notice that if $w_{ij} = 1$ then this is just the gradient covariance of the image and the Harris detector is essentially a local Principle Components Analysis (PCA). In fact Harris doesn't actually compute the eigenvalues but instead a related quantity called the "strength":

$$S = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 \tag{7}$$
$$= \det(M) - \kappa\,\mathrm{trace}^2(M) \tag{8}$$

Hence Bentoutou *et al.* first compute a gradient map of the image using a first order Gaussian derivative filter. They then threshold[1] the gradient map at the average gradient value, thereby extracting only sufficiently "interesting" regions, and compute the strength S for all pixels. They also apply Non-maximum Suppression[2] (NMS) using a 3×3 window and further threshold the remaining non-zero strength values at a threshold of 1% of maximum observed strength. Finally only the "strongest" n corners are kept.

[1] Set everything below a threshold to zero.
[2] Pick the maximum in a neighborhood and set all other values to zero.

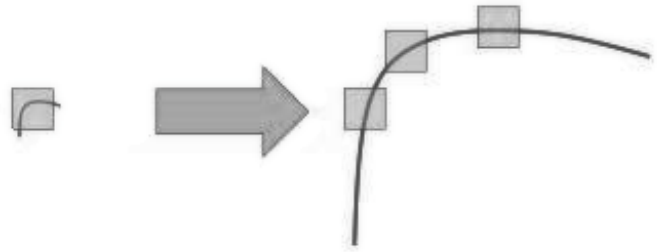


Fig. 3: Harris Detector failing to recognize the right image as a corner.

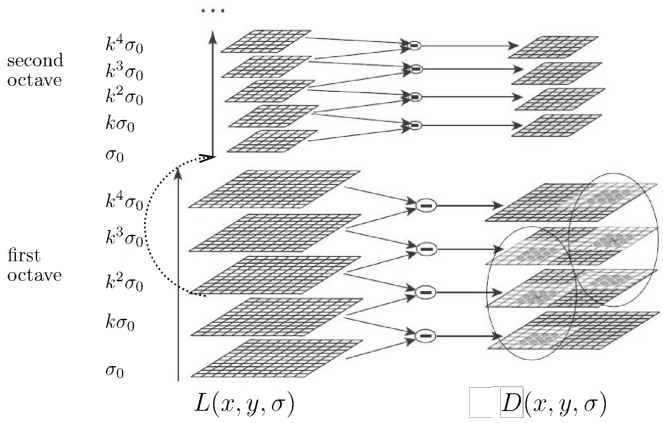


Fig. 4: Smoothed $L(x, y, \sigma)$ and differenced $D(x, y, \sigma)$ pyramid of progressive feature scales**lowe2004distinctive**.

1.1.1.2 SIFT

One issue with Harris detectors is that they're not invariant to scale (see figure **??**). Zahra *et al.***zahrasift** resolves this issue by using the Scale Invariant Feature Transform**lowe2004distinctive** (SIFT) to identify CPs that, as the name implies, are invariant across multiple scales. SIFT identifies scale invariant and noise robust features of an image, called *keypoints*, by first finding candidate points with high local curvature at multiple scales and then culling according to some heuristics.

It then "describes" these keypoints by a rotation invariant and noise robust representation. The algorithm consists of five steps:

1) Scale-space pyramid construction: a sequence of increasingly sub-sampled and more strongly Gaussian filtered images is computed. The sequence of differences of these images is also computed; the sequence of differenced images approximates the multi-scale Laplacian of Gaussians[3] (LoG) of the image (see figure **??**).
2) Keypoint detection: candidate keypoints are points on edges with curvature, i.e. extrema along scale and space dimensions in the LoG pyramid (see figure **??**).
3) Keypoint selection: candidate keypoints are more precisely localized using an iterative process. Keypoints of low-contrast (therefore sensitive to noise) or on edges of low curvature[4] are culled.
4) Keypoint orientation assignment: orientation is assigned to each keypoint by taking a weighted majority vote of all gradient orientations in a neighborhood of the keypoint (see figure **??**a). Large minority votes (80% of majority) are used to create more keypoints at the same pixel point.
5) Keypoint descriptor computation: for each keypoint the descriptor is computed by partitioning the keypoint's neighborhood into 2^k sub-neighborhoods, computing an 8-bin histogram of oriented gradients[5] (HOG) in each sub-neighborhood, and concatenating (see figure **??**b). In Lowe *et al.***lowe2004distinctive** $2^4 = 16$ sub-neighborhoods are used to produce an $8 \times 16 = 128$ entry length descriptor. The descriptor is also normalized to unit length in order to make it invariant to luminance (intensity).

SIFT is indeed effective as a CP detector but unfortunately it is patented. Alternatives include Binary Robust Invariant Scalable Keypoints**leutenegger2011brisk**, and Oriented FAST and rotated BRIEF**rublee2011orb** (which itself consists of applying Features from accelerated segment test**rosten2006machine** to detect points of interest and Binary Robust Independent Elementary Features**calonder2010brief** to compute descriptors).

1.1.2 Feature Matching

After robust features are identified in the reference image and the displaced images they need to be matched. For example for SIFT, where the descriptors are designed to be essentially invariant across images, Euclidean distance using a k-d tree[6] can be used to efficiently match keypoint descriptors. Although this often leads to false-positive matches (Zahra

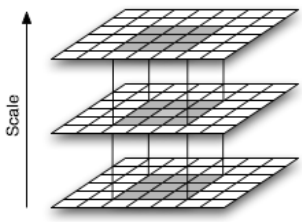


Fig. 5: SIFT local extrema in scale and space.

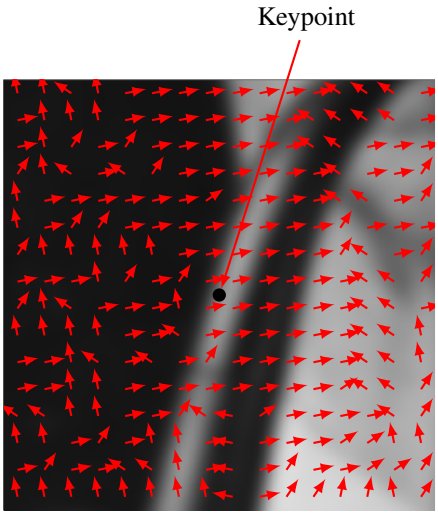
[3]The Laplacian of an image X is $L(x, y) = \partial_x^2 X + \partial_y^2 X$. Since in practice this approximates a noisy signal (second derivative), smoothing by a Gaussian is a necessary prerequisite. Therefore the Laplace of Gaussians (LoG) filter is the Laplacian composed with the Gaussian:

$$LoG(x, y, \sigma) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\}\right]$$

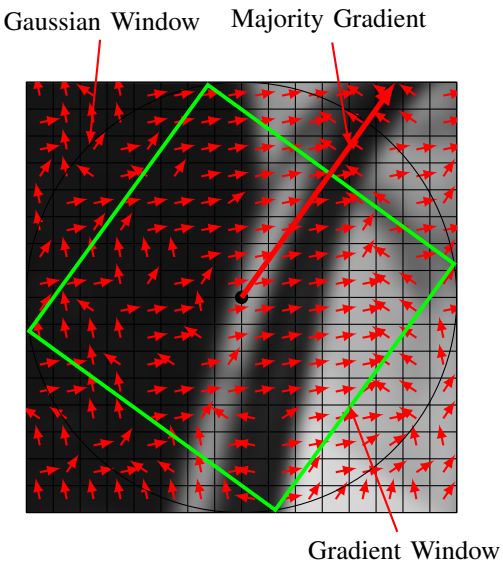
[4]The ratio of the eigenvalues of the spatial Hessian of D (i.e. only along dimensions x, y). In fact a quantity not unlike eqn. (8) is computed in order to save having to explicitly find eigenvalues.

[5]Simply the histogram of gradient orientations (polar angle) in a pixel window.

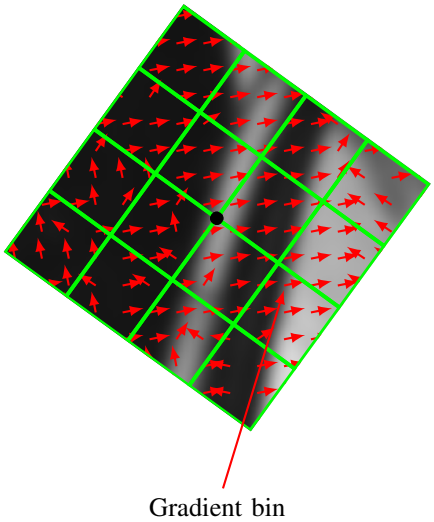
[6]k-d trees**Bentley:1975:MBS:361002.361007**, short for k-dimensional trees, are data structures that partition space efficiently in order that searching the tree, insertion into the tree, and deletion from the tree are all, on average, $O(\log n)$ time operations (where n is the number of nodes in the tree at the time of the operation).



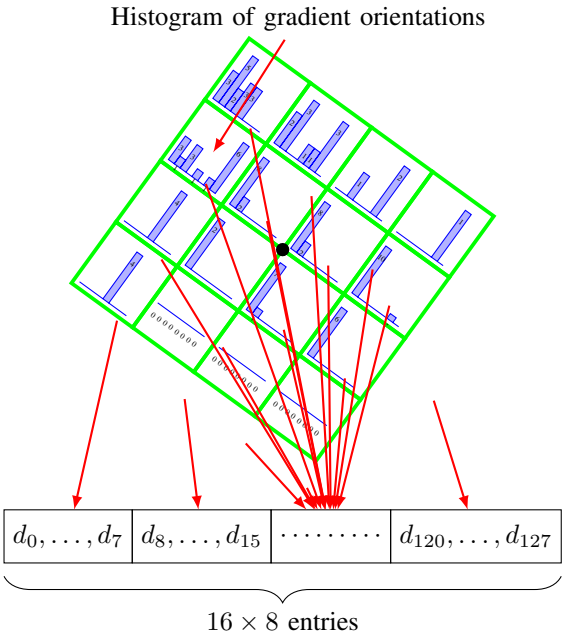
(6.a) Keypoint neighborhood



(6.b) Oriented and filterd keypoint gradient neigbhorhood



(6.c) 16 Keypoint neigbhorhood gradient bins



(6.d) Histogram of gradient orientations descriptor

et al. resolve this by using Random Sample Consensus (RANSAC)[7]) it's a natural feature matching method. In other cases the matching mechanism is not so straightforward; for a class of algorithms called area-based or intensity-based, that in fact combine the feature detection and matching step into one, matching involves comparing summaries of patches in the reference image and the displaced image.

[7]RANSAC is a method used to estimate parameters of a model given outliers. In this case Zahra *et al.* use RANSAC at the transform estimation step to eliminate falsely matched keypoint pairs. RANSAC iterates by repeatedly random sampleing the putative matching keypoint pairs and fitting a transform model. At a given iteration the fitted transform model is tested against the unused keypoint pairs and evaluated (according to goodness of fit on a subset called the *consensus set*).

1.1.2.1 Normalized Cross-Correlation

1.1.3 Transform Estimation

Furthermore, the transformation to be estimated should incorporate prior knowledge about the motion model but simultaneously lead to a tractable estimation problem (i.e. reasonable number of parameters).

1.2 Gaussian Process