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. The estimation process can be broken down into three distinct steps: feature detection, feature matching, and mapping function estimation.

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 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:

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

where $x_i := x + i$ and $y_j := y + j$. 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:

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

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

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

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

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

$$= [u, v] \, M \begin{bmatrix} u \\ v \end{bmatrix} \qquad (6)$$

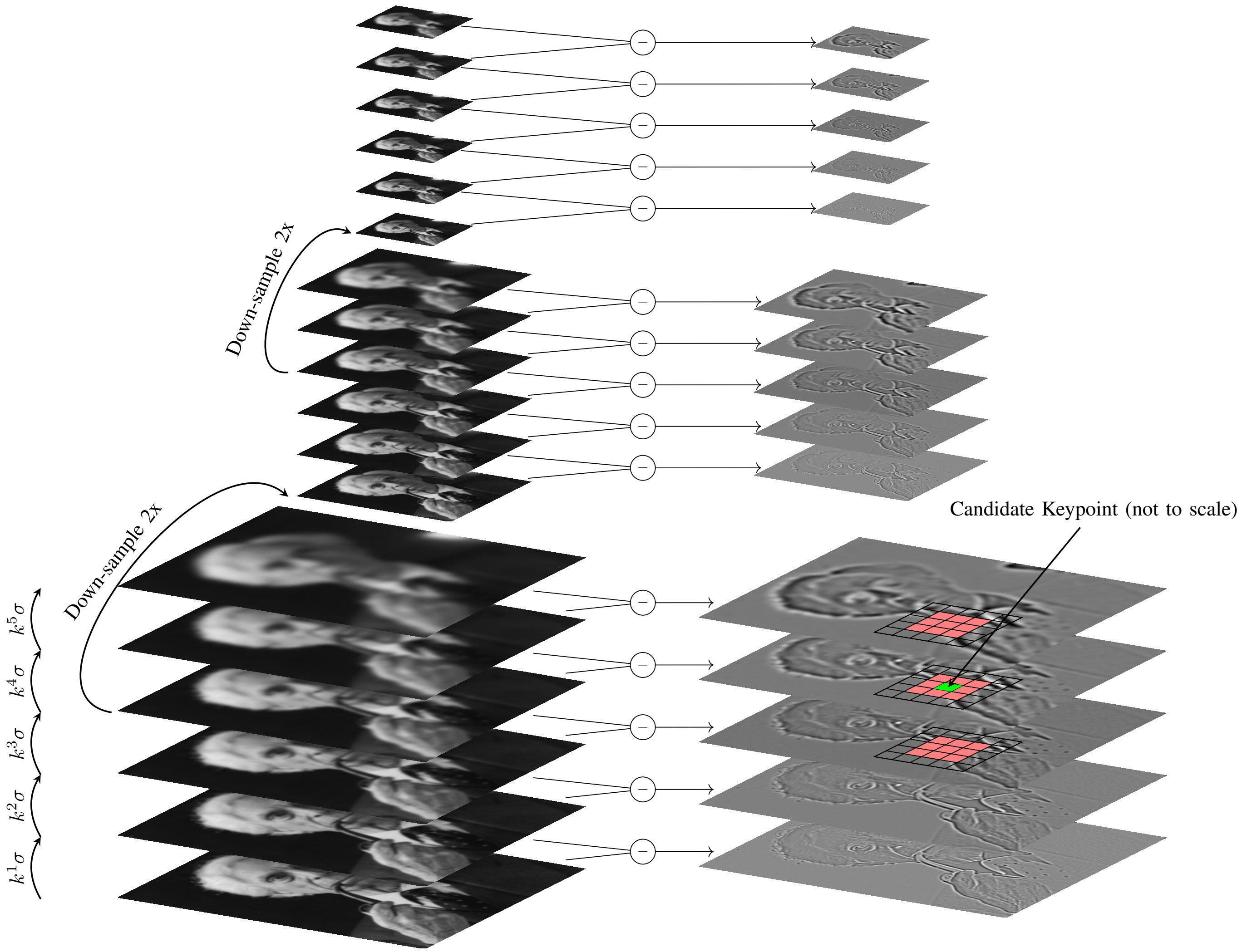


Fig. 1: Gaussian filtered and sub-sampled pyramid and corresponding difference of Gaussians pyramid for identifying keypoints (as local gradient maxima).

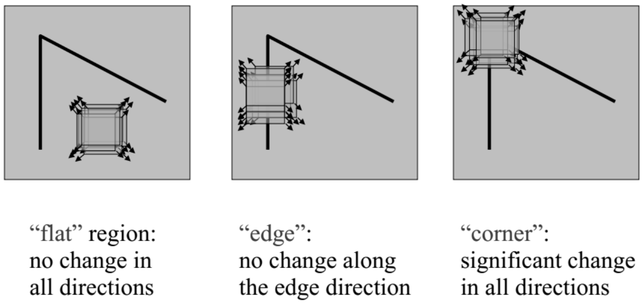

"flat" region: no change in all directions
"edge": no change along the edge direction
"corner": significant change in all directions

Fig. 2: Moravec Corner Detector

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 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":

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

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

[1] Set everything below a threshold to zero.

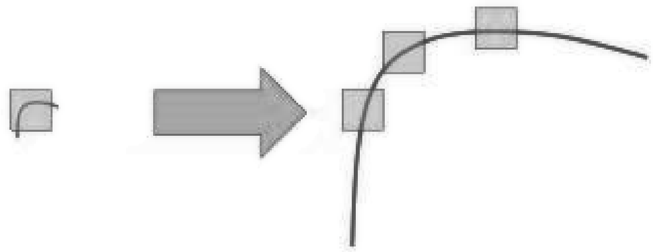


Fig. 3: Harris Detector failing to recognize the right image as a corner.

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.

1.1.2 SIFT

One issue with Harris detectors is that they're not invariant to scale (see figure 3). 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 1).
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 1).
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 4b). 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 4d). 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).

1.2 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 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 *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 algorithms, 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.

1.2.1 Convex Hull Edges

One technique that matches features explicitly is Point Matching Using Convex Hull Edges**Goshtasby1985**. It operates on the principle that the convex hull[8] of a set of points is invariant under translation, rotation, and scaling. Goshtasby *et al.* further argue that a noised set of points will most likely

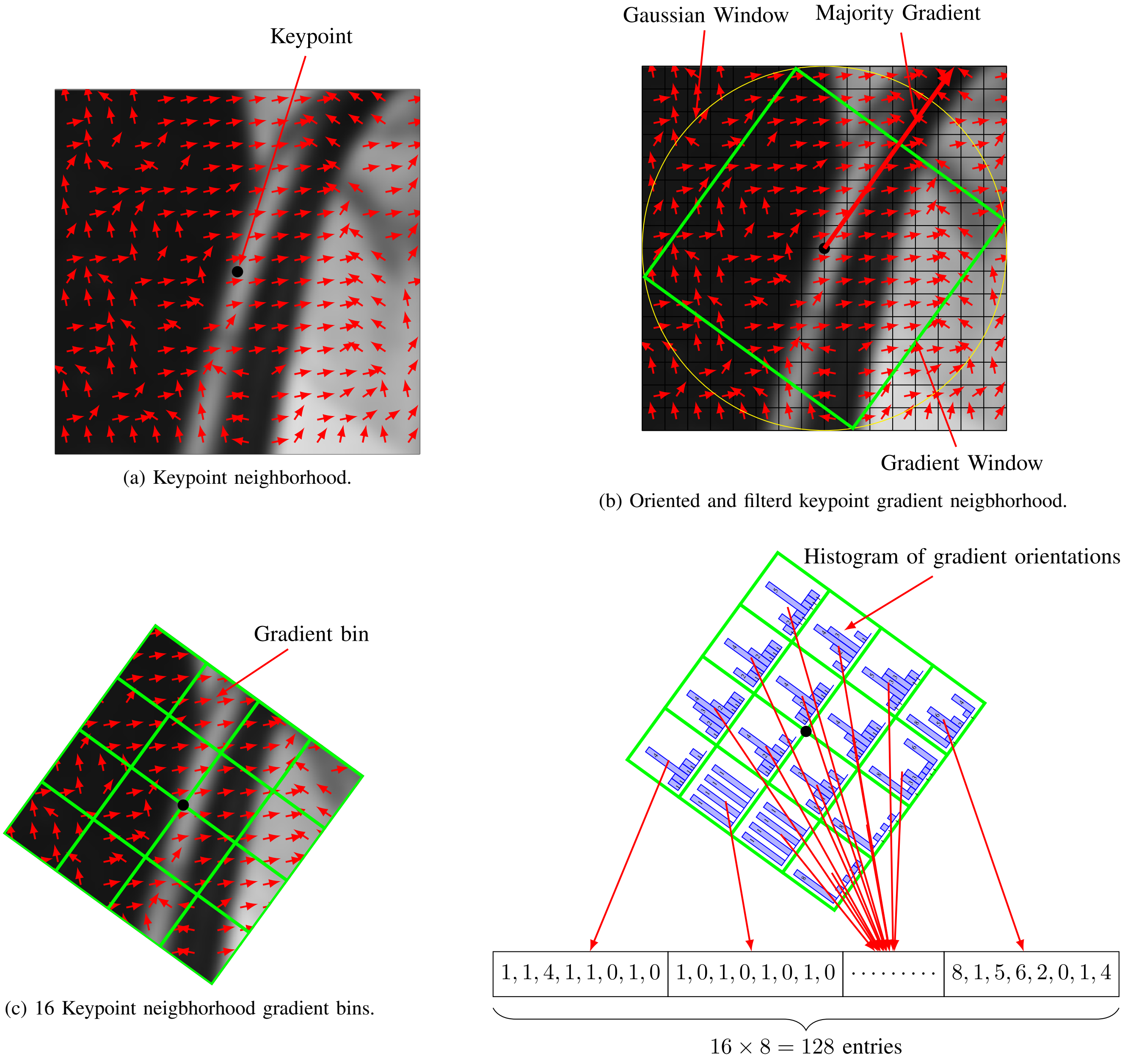
[2] Pick the maximum in a neighborhood and set all other values to zero.

[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 data structure, insertion into the data structure, and deletion from the data structure 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).

[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 sampling 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*).

[8] A convex combination of points $x_1, x_2, \dots, x_n$ is all $\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$ where $\alpha_i \geq 0$ and $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$. The convex hull of a set of points X is the set of all convex combinations of points in X. Schematically one can imagine stretching a rubberband such that it contains all points in X.

;



(a) Keypoint neighborhood.

(b) Oriented and filterd keypoint gradient neigbhorhood.

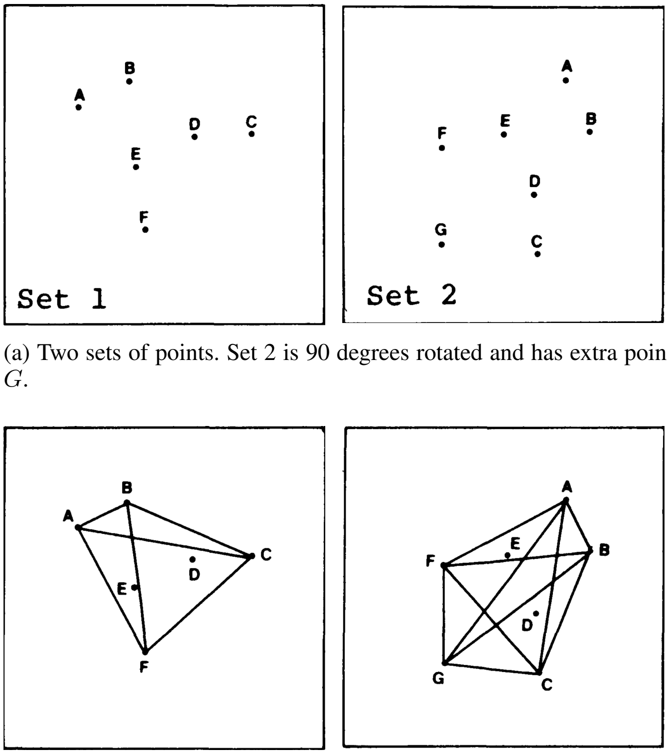
(c) 16 Keypoint neigbhorhood gradient bins.

(d) Histogram of gradient orientations descriptor.

Fig. 4: Sift keypoint descriptor construction.

have elements deleted or added in its interior and therefore register the boundaries of the respective convex hulls of the to-be-registered image and the reference image (see figure 5). The key part of the algorithm (after computing the convex hull of the control points) is estimating transformation parameters between convex hull boundaries C_1, C_2:

1) Determine the complete graphs[9] $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ of C_1 and C_2. Then
$$|E_1| = \frac{|V_1|(|V_1| - 1)}{2}$$
($|V_1|$ "choose" 2) and similarly $|E_2|$. Since there are 2 ways to match any undirected edge we extend E_2 to include reverse directed edges and therefore
$$|E_2| = |V_2|(|V_2| - 1)$$
2) For each possible pairing of edges e_i, e_j in E_1, E_2 compute the rotation R_{ij}, scaling S_{ij}, and translation T_{ij}.
3) Given the composite transformation $R_{ij} \circ S_{ij} \circ T_{ij}$ count the number N_{ij} of other points in C_1, C_2 that also match (within a threshold distance D).
4) $N_{IJ} = \max_{i,j} N_{ij}$ and $R_{IJ} \circ S_{IJ} \circ T_{IJ}$ are the transformation parameters between the control points.



(a) Two sets of points. Set 2 is 90 degrees rotated and has extra point G.

(b) Convex hull complete graph edges of two sets, used by the point matching using convex hull edges algorithm.

Fig. 5: Point matching using convex hull edges**Goshtasby1985**.

Note that this procedure operates on one set of control points at a time and that there could be many such matchings computed before final registration between images can be performed.

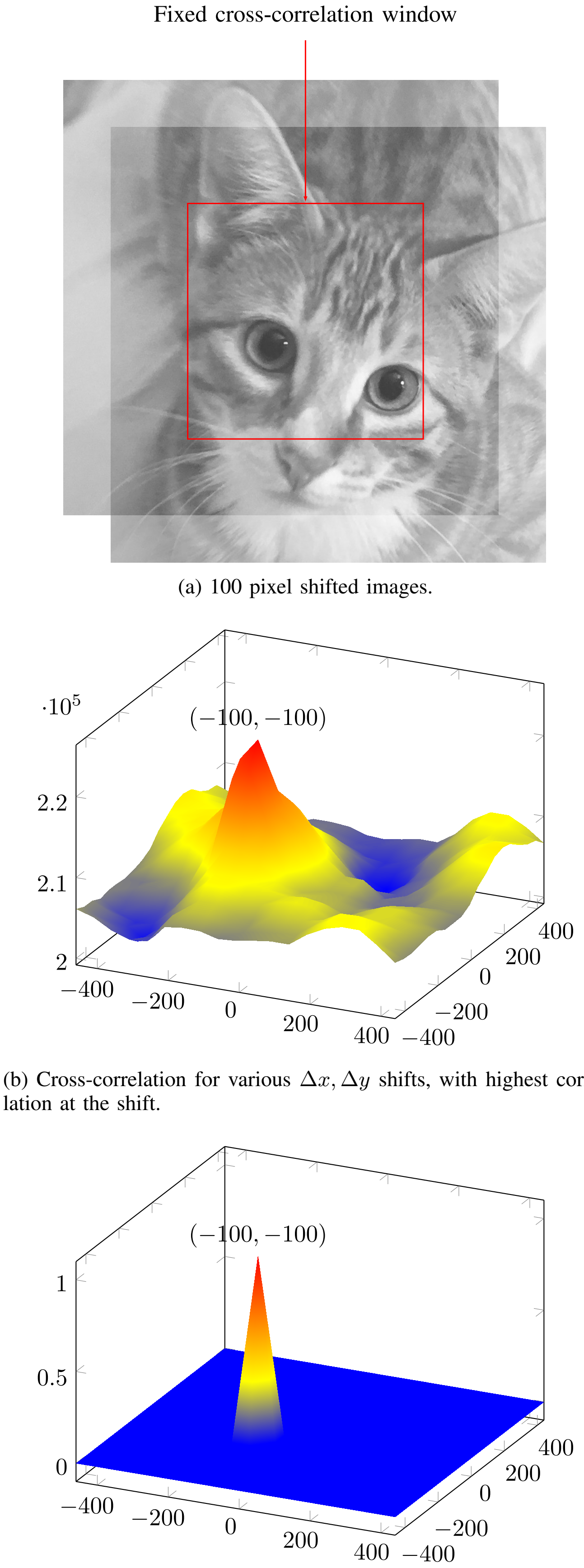
1.2.2 *Normalized Cross-correlation*

The simplest strategy for matching by region is to grid-search[10] the space of possible translational shifts $\Delta x, \Delta y$ and assess the quality of the registration for a given shift using a similarity metric. One such similarity metric is Normalized Cross-correlation (NCC); for two image patches X, Y (with the same width, height) NCC is defined in a straightforward way

$$NCC(X,Y) = \frac{\sum_{x,y}\left(X(x,y)-\hat{X}\right)\left(Y(x,y)-\hat{Y}\right)}{\sqrt{\sum_{x,y}\left(X(x,y)-\hat{X}\right)^2}\sqrt{\sum_{x,y}\left(Y(x,y)-\hat{Y}\right)^2}} \tag{9}$$

where $\hat{X}, \hat{Y}$ are mean patch values.

The NCC image registration technique performs a grid-search over possible shifts and compute the cross-correlation of the shifted images (see figure 6a) (usually over a fixed cross-correlation window rather than the entire image for the sake of computational efficiency). The maximum response as a



(a) 100 pixel shifted images.

(b) Cross-correlation for various $\Delta x, \Delta y$ shifts, with highest correlation at the shift.

(c) Phase-correlation for various $\Delta x, \Delta y$ shifts, with single peak at the correct shift.

Fig. 6: Cross-correlation feature matching.

[10] Searching a space by taking regular incremental steps along each dimension (e.g. search $U \subset \mathbb{R}^2$ by checking all $x = \Delta x + x_0, y = \Delta y + y_0$ for $(x_0, y_0) \in U$ and for fixed $\Delta x, \Delta y$).

function $\Delta x, \Delta y$ is the imputed translation between the images (see figure 6b). An alternative but closely related method is *phase correlation*, based on the Fourier shift theorem[11] (the value of going to Fourier space is availability of highly optimized algorithms for computing the Fourier transform). Let $\mathcal{X}(u,v) = \mathcal{F}\{X(x,y)\},\ \mathcal{Y}(u,v) = \mathcal{F}\{Y(x,y)\}$ be the Fourier transforms of the displaced images. Then

$$\mathcal{X} = \mathcal{Y} e^{-2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}$$

where N, M are the dimensions of the images and the *normalized cross-power spectrum*

$$\begin{aligned} R(u,v) &= \frac{\mathcal{X} \circ \mathcal{Y}^*}{|\mathcal{X}||\mathcal{Y}^*|} && (10) \\ &= \frac{\mathcal{X} \circ \mathcal{X}^*\, e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}}{|\mathcal{X}||\mathcal{X}^*\, e^{-2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}|} && (11) \\ &= \frac{\mathcal{X} \circ \mathcal{X}^*\, e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})}}{|\mathcal{X}||\mathcal{X}^*|} && (12) \\ &= e^{2\pi i (\frac{u\Delta x}{M} + \frac{v\Delta y}{N})} && (13) \end{aligned}$$

where $\circ$ is Hadamard product[12], $|\mathcal{X}|$ is the magnitude of $\mathcal{X}$, $\mathcal{X}^*$ is the complex conjugate of $\mathcal{X}$ and we've used the fact that $|e^{iz}| = 1$ for all z. Then the inverse Fourier transform of eqn. (13)

$$\mathcal{F}^{-1}\{R(u,v)\} = r(x,y) = \delta(x + \Delta x, y + \Delta y)$$

is a single peak (see figure 6c) at $(\Delta x, \Delta y)$. The simplest implementation of NCC only identifies translations but it can be extended to affine transforms**berthilsson1998**.

1.2.3 Mutual Information

In general NCC fails for noisy images (i.e. the to-be-registered image is much noisier than the reference image). Mutual information (MI) methods have been successfully employed in such cases; MI is also robust to changes in light intensity, pixel color and can fit large transformations. The mutual information of two random variables X_1, X_2 is defined

$$I(X_1, X_2) := H(X_1) + H(X_2) - H(X_1, X_2)$$

where H is Shannon-entropy[13]. In this context X_1 is reference image and X_2 is the to-be-registered image. The central challenge in computing MI is in estimating joint $H(X_1, X_2)$ and marginal $H(X_1)$ entropies (which are functions of the joint $P(X_1, X_2)$ and marginal $P(X_1)$ probabilities). In practice this is done using either Kernel Density Estimation[14] (KDE) or the joint histogram (i.e. number of common pixel values at same pixel coordinates normalized by total number of pixels). Searching for the registration parameters θ that maximize $I_\theta := I(X_1, X_2(\theta))$ exhaustively is generally costly (especially if more than just translation parameters are required). Ritter *et al.***ritter1999** use simulated annealing to efficiently search the space of all possible parameters. Simulated annealing iterates on the current candidate minimum I_θ by "jumping" to potentially new minima in a way that prioritizes near jumps but allows for the possibility of far jumps. They use the Cauchy distribution as the jump proposal function:

$$G_k(d) = \frac{t_k}{\pi\,(d^2 + t_k^2)}$$

where t_k is a parameter that over time encourages near jumps more and more strongly (called the temperature), and d is the parameter jump distance. To allow escape from local minima of I_θ it is necessary to occasionally make a jump from the current parameter estimate θ_k to a proposed value θ_{k+1} that is "worse" (i.e. for which $I(\theta_{k+1}) < I(\theta_k)$). This is accomplished by accepting or rejecting the proposal with probability determined by an *acceptance function* A_k:

$$A_k(\theta_k, \theta_{k+1}) = \begin{cases} 1 & \text{if } I(\theta_{k+1}) \geq I(\theta_k) \\ \frac{1}{Ct_k} e^{I(\theta_k) - I(\theta_{k+1})} & \text{otherwise} \end{cases}$$

1.3 Transform Estimation

After feature matching is sucessfully completed the transformation function f is constructed (see eqn. (**??**)). This consists of choosing a transformation type (see figure 7) and estimating the parameters necessary to uniquely determine the transformation. The transformation type is partially informed by knowledge of the geometry of the displacements (rigid, affine, projective) and partially informed by computational efficiency (more general transformation have more parameters). There are broadly two perspectives on constructing the transformation f: 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. We first quickly cover global algorithms and then move on to more powerful, and more interesting, local algorithms.

1.3.1 Global Algorithms

The most commonly used global model is the rigid transformation (see figures 7a), which in homogeneous coordinates[15] is defined:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (14)$$

[11] If $\mathcal{F}\{[x_n]\}_k = X_k$ is the kth frequency component of the discrete signal $[x_n]$ then shifting $[x_n]$ by m steps implies

$$\mathcal{F}\{[x_{n-m}]\}_k = X_k \cdot e^{-\frac{i2\pi}{N}km}$$

[12]

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \circ \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} := \begin{bmatrix} a_{11}\,b_{11} & a_{12}\,b_{12} \\ a_{21}\,b_{21} & a_{22}\,b_{22} \end{bmatrix}$$

[13] Shannon entropy can be interpreted as the number of bits L necessary to encode a random variable X distributed according to distribution P. It is defined as

$$\mathrm{H}(X) := \sum P(x) \log\left[\frac{1}{P(x)}\right]$$

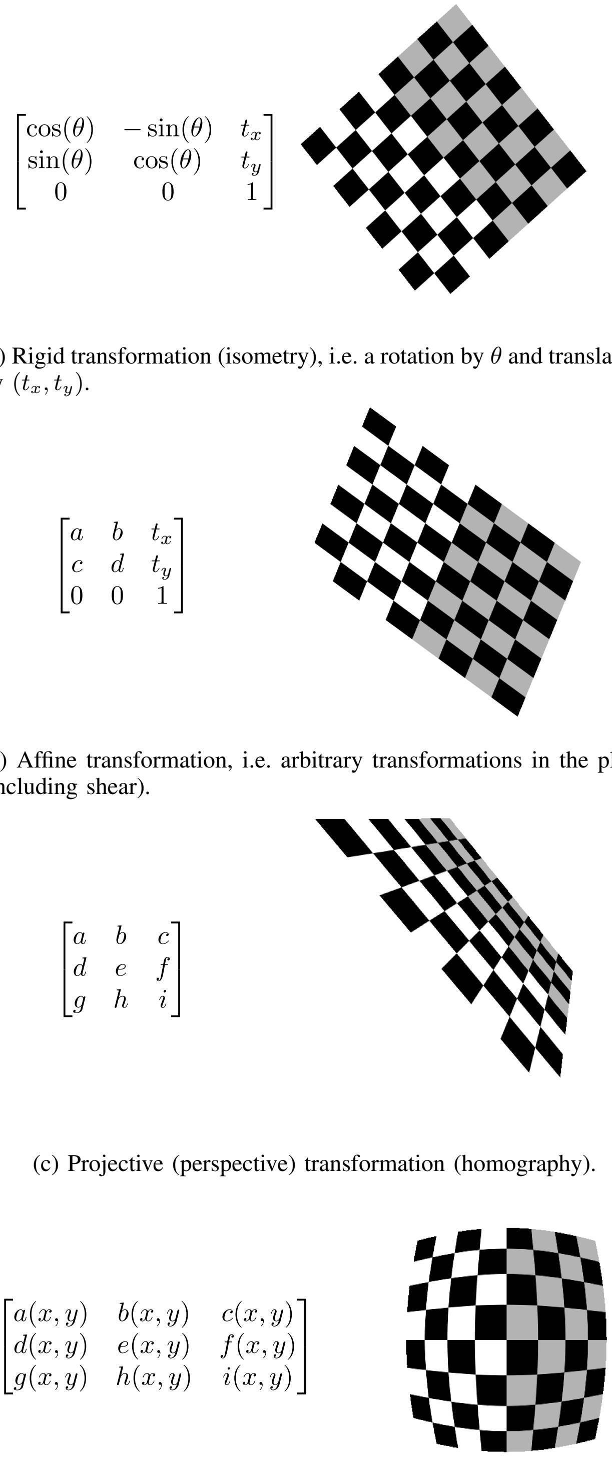
i.e. $P(x_i) = 1/2^L$ where L is the number of bits.

[14] Alternatively known as Parzen windowing, the kernel density estimator $\hat{P}$ of a distribution $X \sim P$ given a set of points $\{x_1, x_2, \dots, x_n\}$ is defined as

$$\hat{P}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where K is a kernel (e.g. Gaussian) and h is the *kernel bandwidth*.

[15] With (x,y) a pixel coordinate, transformations in homogeneous coordinates operate on $x, y, 1$: if $(a,b,c) = (x,y,1)\,T$ then the transformed pixel coordinate $(x', y') = \frac{1}{c}(a,b)$.



$$\begin{bmatrix}\cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1\end{bmatrix}$$

(a) Rigid transformation (isometry), i.e. a rotation by θ and translation by (t_x, t_y).

$$\begin{bmatrix}a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1\end{bmatrix}$$

(b) Affine transformation, i.e. arbitrary transformations in the plane (including shear).

$$\begin{bmatrix}a & b & c \\ d & e & f \\ g & h & i\end{bmatrix}$$

(c) Projective (perspective) transformation (homography).

$$\begin{bmatrix}a(x,y) & b(x,y) & c(x,y) \\ d(x,y) & e(x,y) & f(x,y) \\ g(x,y) & h(x,y) & i(x,y)\end{bmatrix}$$

(d) Elastic transformation, i.e. locally defined transformation.

Fig. 7: Different transformation classes (in homogeneous coordinates) along with realized examples.

The unknown parameters (θ, t_x, t_y) in eqn. (14) can be estimated by expanding x', y' to using Taylor series

$$\begin{aligned} x' &= t_x + x\left(1 - \frac{\theta^2}{2}\right) - y\theta + O(\theta^3) \\ y' &= t_y + x\theta + y\left(1 - \frac{\theta^2}{2}\right) + O(\theta^3) \end{aligned} \tag{15}$$

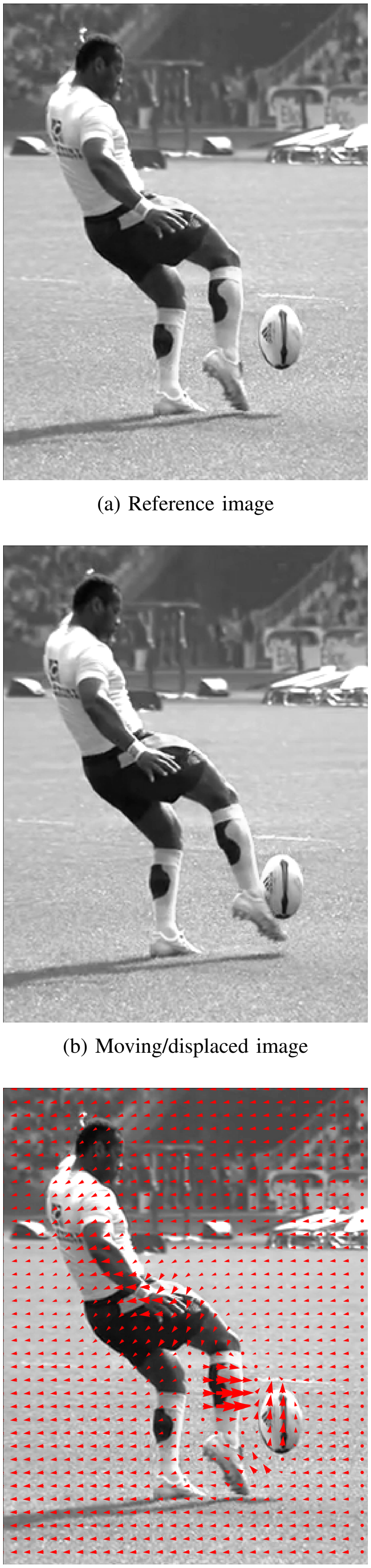
Note that this is only valid for small angles θ (the small-angle approximation**keren1988**). Truncating eqns. (15) to first or second order and substituting data from matched features yields a system of linear equations. Given that often there are many more matched features than unknowns in eqns. (15) such a system is overdetermined; a unique solution can be determined using regularized least-squares. In order to make this process more efficient and increase accuracy Keren *et al.***keren1988** use a Gaussian pyramid (see figure 1 and corresponding discussion in section 1.1.2) to iteratively refine estimates at higher and higher resolutions. In the more general case of a projective transformation (a perspective shift) the transformed pixel coordinates are

$$\begin{aligned} x' &= \frac{ax + by + c}{gx + hy + i} \\ y' &= \frac{dx + ey + f}{gx + hy + i} \end{aligned} \tag{16}$$

and similarly the unknowns $(a, b, \ldots$, etc.) can be estimated from matched feature correspondence data.

1.3.2 Local Algorithms

The most general and most interesting framing of the problem of local transformation is optical flow. Optical flow is the apparent motion of objects, surfaces, and edges in a scene caused by the relative motion between an observer and a scene. It is most often represented as a velocity field over the individual pixels (see figure 8).



(a) Reference image

(b) Moving/displaced image

(c) Velocity field

Fig. 8: Optical flow