LiDAR-to-Optical Registration

William L. Ruys

September 2020

1 Introduction

2 Methods

2.1 Fourier Based Registration

Suppose you have two gray-scale image functions $I_1,\ I_2 : \mathbf{x} \in R^2 \mapsto [0, 255]$. Without loss of generality assume the images have the same width and height, N. If this is not the case, they can be zero padded to meet this condition.
Let I_2 be a shift of I_1 such that $I_2(\mathbf{x}) = I_1(\mathbf{x} + \mathbf{t})$. Then this translation $\mathbf{t}$ can be estimated as the shift that gives the maximum correlation between the two images. The naive evaluation of this takes $O(|T|N^2)$ time to evaluate for all translations in a prospective set T. If the correlation function can be expressed as a linear combination of convolutions this calculation can be accelerated through the Fourier transform. This allows us to evaluate the correlation for $\mathbf{t} \in T = \{[-\frac{N}{2}, \frac{N}{2}] \times [-\frac{N}{2}, \frac{N}{2}]\}$ in $O(N^2 log(N))$ time. We consider a few correlation functions in our Python implementation that we'll outline briefly.

Throughout this writeup, let $\hat{f} = \mathcal{F}(f)$, its Fourier transform. In a gross abuse of theory and notation we also overload this hat notation to mean the discrete Fourier transform in the algorithms below. Let $\nabla I_i = \nabla_x I_i + j\nabla_y I_i$, where $j = \sqrt{-1}$.

2.1.1 Cross-Correlation Function $\mathbf{C}(\tau)$

$$\begin{aligned} C(\tau) &= \int_{R^2} I_1(\mathbf{x}) I_2(\mathbf{x} + \tau) d\tau \\ &= \mathcal{F}^{-1}[\hat{I_1}(k)\hat{I_2}^*(k)] \end{aligned}$$

2.1.2 Phase Correlation (Cross-Power) $\mathbf{PC}(\tau)$

$$PC(\tau) = \mathcal{F}^{-1}[\frac{\hat{I_1}(k)\hat{I_2}^*(k)}{|I_1(k)||I_2^*(k)|}]$$

2.1.3 Normalized Gradient Correlation NGC(τ)

$$
\begin{aligned}
NGC(\tau) &= \frac{\int_{R^2} \nabla I_1(\mathbf{x}) \nabla I_2(\mathbf{x}+\tau) d\tau}{\int_{R^2} |\nabla I_1(\mathbf{x})||\nabla I_2(\mathbf{x}| + \tau) d\tau} \\
&= \frac{\mathcal{F}^{-1}[\hat{\nabla I_1}(k) \hat{\nabla I_2}^*(k)]}{\mathcal{F}^{-1}[|\hat{\nabla I_1}|(k) |\hat{\nabla I_2}|^*(k)]}
\end{aligned}
$$

2.1.4 Normalized Graident Fields NGF(τ)

Let $n_x(I_i) = \frac{\nabla_x I_i}{|\nabla I_i|}$, $n(I_i) = [n_x(I_i), n_y(I_i)]$.

$$
\begin{aligned}
NGF(\tau) &= \int_{R^2} (n(I_1(\mathbf{x})) \cdot n(\nabla I_2(\mathbf{x}+\tau)))^2 d\tau \\
&= \int_{R^2} n_x^2(I_1(\mathbf{x})) n_x^2(\nabla I_2(\mathbf{x}+\tau)) d\tau + \int_{R^2} n_y^2(I_1(\mathbf{x})) n_y^2(\nabla I_2(\mathbf{x}+\tau)) d\tau \\
&+ \int_{R^2} (n_x \cdot n_y)(I_1(\mathbf{x}))(n_x \cdot n_y)(\nabla I_2(\mathbf{x}+\tau)) d\tau
\end{aligned}
$$

2.1.5 Log-Log Interpolation

Consider I_2 as an anisotropic scaling and translation of I_1 such that $I_2(\mathbf{x}) = I_1(S\mathbf{x} + \mathbf{t})$ where $S = \text{Diag}([s_1, s_2])$. The translation can be separated from the scaling by taking a Fourier transformation, where the relationship between the two images is $\hat{I}_2(\mathbf{k}) = \frac{1}{|S|} \hat{I}_1(S^{-1}\mathbf{k}) e^{j\mathbf{t}}$. If we take the magnitude (complex modulus) the phase component is dropped and we just have a scaling (with adjusted amplitude), $M_2(k) = \frac{1}{|S|} \hat{M}_1(S^{-1}k)$.

We can change a scaling to a translation by performing a trick where we change the spacing of the interpolation points to a logarithmic scale (instead of linear) and view the function as if these interpolation points were linearly spaced. Under this interpolation $\tilde{M}_2(k_1, k_2) \propto \tilde{M}_1(k_1 - \log(s_1), k_2 - \log(s_2))$. These scaling parameters can then be found by the correlation method described above. Once the I_2 has been warped to the found rescaling, only a translation remains between the two images. This can be computed via a second pass of the correlation method.

Result: Estimate of Affine Warping Parameters: ($\mathbf{t}$, $\tilde{S_w}, \tilde{S_h}$)
Data: Image fuctions: I_1, I_2 related by scaling and translation
Data: Oversampling parameter l
$b_i = (\frac{N_i}{2})^{\frac{1}{lN_i}}$, $N_i = I_2$.shape[i];
Apply windowing function to I_1, I_2;
Zero pad I_1, I_2;
Take 2D FFT of ∇I_1 and ∇I_2;
$M_1, M_2 =$ magnitudes of $\hat{\nabla} I_1$ and $\hat{\nabla} I_2$;
Interpolate M_1 and M_2 to log-log coordinate system with bases (b_1, b_2);
Apply windowing function to M_1, M_2;
Zero pad M_1, M_2;
Take 2D FFT of ∇M_1 and ∇M_2;
Compute element-wise product, $\frac{\hat{\nabla M_1}(k)\hat{\nabla M_2^*}(k)}{|\hat{\nabla M_1}(k)||\hat{\nabla M_2^*}(k)|}$;
Take 2D iFFT of product to get $PC(\tau)$. $O(N^2 \log(N))$;
$(e_1, e_2) = \arg\max$ of $PC(\tau)$;
$\tilde{S_w} = (b_1)^{e_1}$, $\tilde{S_h} = (b_2)^{e_2}$;
Warp & Interpolate I_2 by $(\tilde{S_w}, \tilde{S_h}$;
Take 2D FFT of I_1 and I_2;
Compute element-wise product, $\frac{\hat{\nabla I_1}(k)\hat{\nabla I_2}^*(k)}{|\hat{\nabla I_1}(k)||\hat{\nabla I_2^*}(k)|}$;
Take 2D iFFT of product to get $PC(\tau)$;
$\mathbf{t} = \arg\max$ of $PC(\tau)$;
Return ($\mathbf{t}$, $\tilde{S_w}, \tilde{S_h}$);

Algorithm 1: Log-Log Fourier Based Registration

2.2 Exhaustive Registration

Result: Estimate of Affine Warping Parameters: $(\tilde{\theta}, \tilde{S_w}, \tilde{S_h})$
Data: Parameter Search Space: $P = (\theta^{1:N_\theta}, S_w^{1:N_{S_W}}, S_h^{1:N_{S_h}})$
Data: Downsampled Images: J_1, J_2
Data: nb = Maximum Batch Size (Determined by GPU Memory)
N = $|P|$;
itr = N/nb;
$J_1 = FFT[\nabla J_1]$;
for *l in 1:itr* **do**
    W = Batch Warp I_2 by subset of P;
    Compute ∇W for each image;
    Batch Compute $W = FFT[\nabla W]$;
    $PC = (W)(J_1^*)/(|W||J_1^*|)$;
    Batch Compute $PC = iFFT[PC]$;
    $(\tilde{\theta}, \tilde{S_w}, \tilde{S_h})$ = argmax(PC, $(\tilde{\theta}, \tilde{S_w}, \tilde{S_h})$);
end

Algorithm 2: Exhaustive Search

Result: Moving Image I_2 aligned at maximum correlation peak

Parameter Search Space: $P = (\theta^{1:N_\theta}, S_w^{1:N_{S_W}}, S_h^{1:N_{S_h}})$;

Downsampling rates: R = [0.1, 0.1, 0.8];

for r *in* R **do**

  J_1, J_2 = Downsample I_1, I_2 by r;

  Pad I_1 and I_2 to (2^M, 2^N), nearest containing powers of 2 ;

  $(\tilde{\theta}, \tilde{S_w}, \tilde{S_h})$ = exhaustiveSearch(J_1, J_2, P);

  Restrict $(\theta^{1:N_\theta}, S_w^{1:N_{S_W}}, S_h^{1:N_{S_h}})$;

end

Warp I_2 by $(\tilde{\theta}, \tilde{S_w}, \tilde{S_h})$;

Perform phaseCorrelation on I_1, I_2

Algorithm 3: Multilevel Exhaustive Search

3 Results

3.1 Exhaustive Registration

3.1.1 Image Recovery

3.1.2 Failure Cases

3.2 Log-Log Fourier Based Registration