CS180 Project 1 · Images of the Russian Empire

Fall 2025

Author: Sammie Smith

Overview

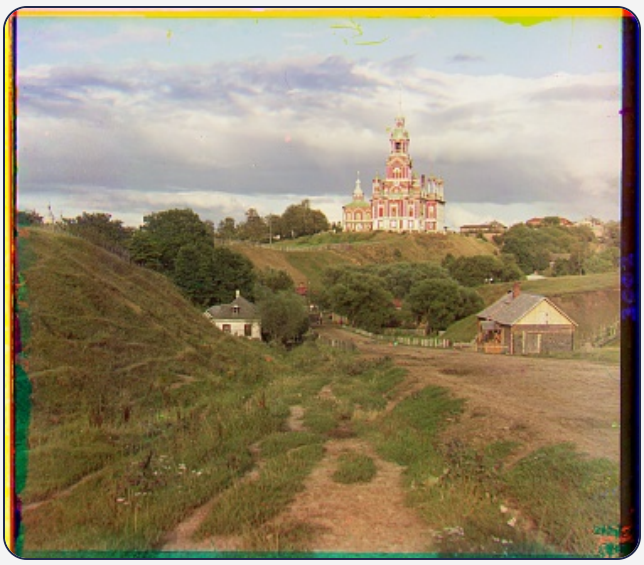
We reconstruct color photographs from Sergei Mikhailovich Prokudin-Gorskii's three filtered glass plates by **auto-aligning** the red and green channels to blue and stacking them into RGB. The code uses robust similarity scoring and a coarse-to-fine search to align quickly and resist lighting variation.

Similarity: `NCC` (or L2) Edges: `Sobel` Border crop: `10 px` Coarse window: `±48`

Refine window: `±15` Max levels: `6` Formats: `uint8/uint16 → [0,1]`

Naïve Method

Split the stacked grayscale plate vertically into **B, G, R**; for each of G and R, perform an exhaustive search over displacements within a small window relative to B. Score candidate shifts with **NCC** (zero-mean, variance-normalized) or **L2 Norm**. Crop borders (naively) to avoid edge artifacts and compare only the **overlapping slices** to prevent out-of-bounds and unfair comparisons. Reconstruct the color image by rolling channels by the chosen offsets and stacking to RGB.



NCC yields sharper alignment than L2 under exposure differences.

L2 is sensitive to brightness/contrast; can blur when channels differ.

Naïve Method Improved

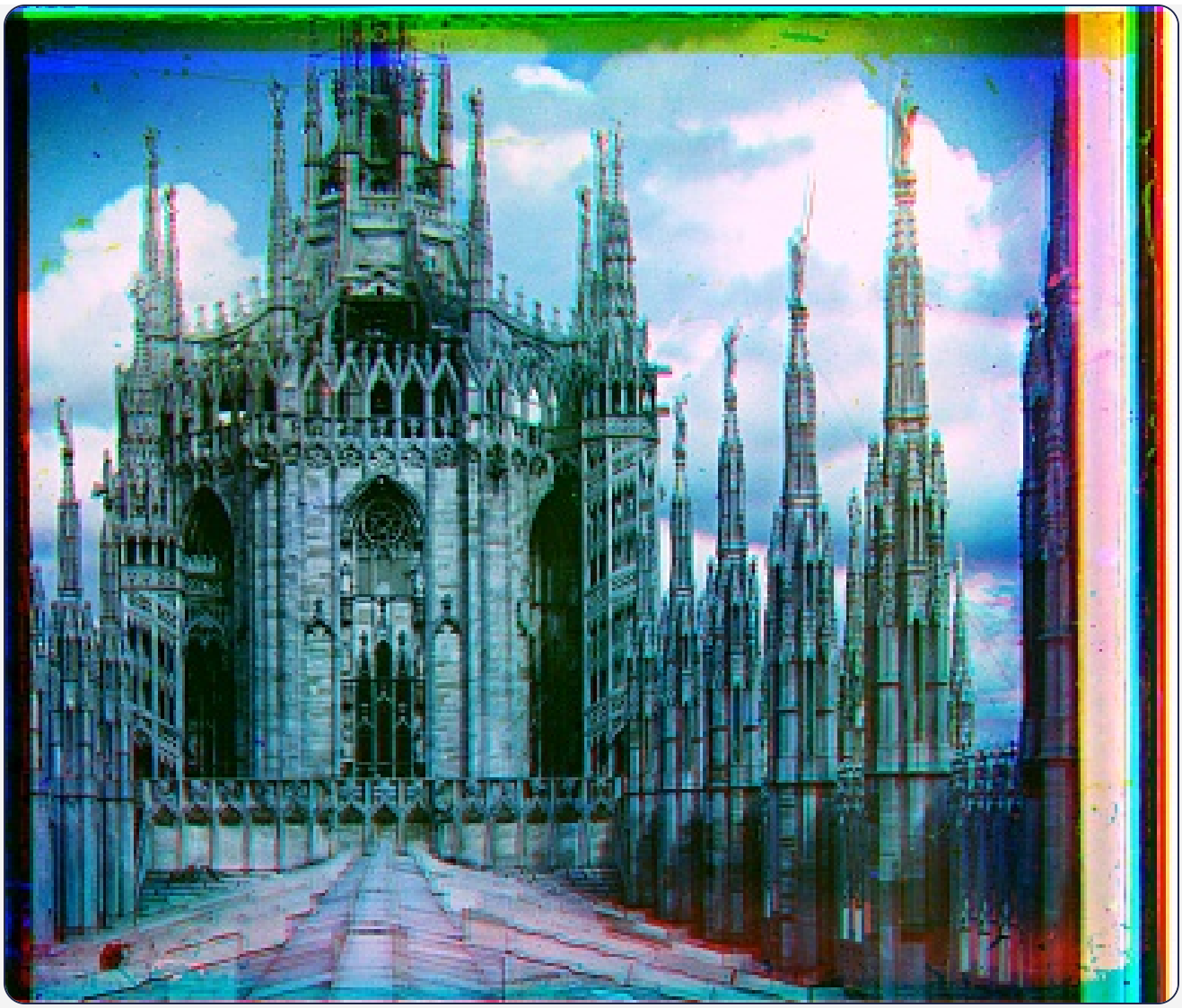
Percentile stretching of channels can reduce outliers but may flatten highlights/shadows if overdone. I chose to increase intensity of green by 1st-99th percentile and blue/red by 10th-90th percentile. This worked for most images, but is not a one-all fix.

NCC + Stretching provides a good color balance here.

NCC + Stretching makes greens too intense.

Image Pyramid Speedup

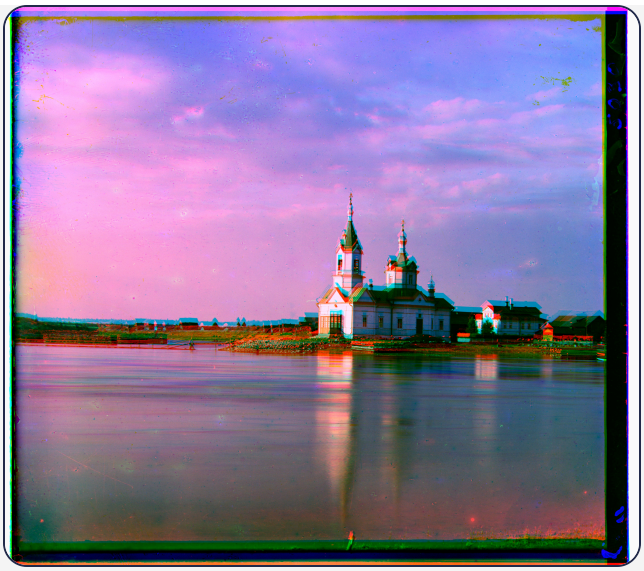
Build a Gaussian pyramid (downsample ×2 per level). At the *coarsest* level, do a wide brute-force search; scale the best shift by 2 and *refine locally* at the next level. Repeat until full resolution. This keeps large searches cheap and high-res refinement focused, delivering big speedups with accurate final offsets.



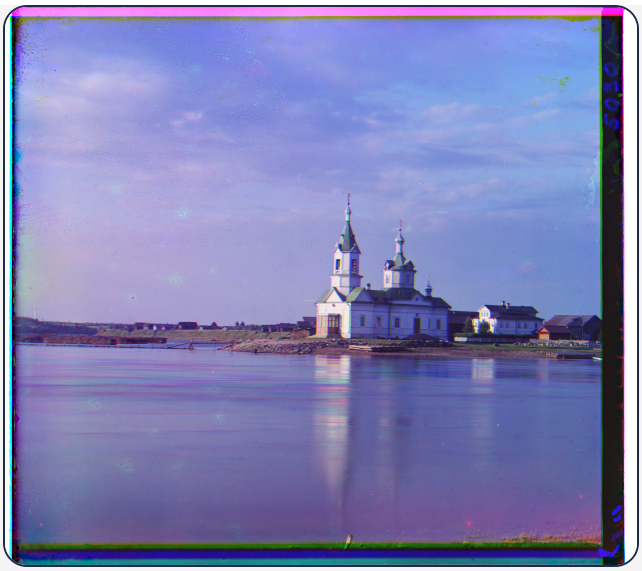
Baseline pyramid alignment (coarse→fine displacement refinement).

Improvements (“Bells & Whistles”)

- **Robustness:** normalize `uint8/uint16` inputs to `[0,1]`; score with **NCC** to ignore brightness/contrast shifts.
- **Feature focus:** compare **Sobel edge** maps rather than raw intensities for lighting-invariant texture matching.
- **Fair comparisons:** compute **overlapping slices** for each shift to avoid padding/garbage pixels and skip tiny overlaps.
- **Border handling:** crop a small interior region to remove bright plate edges that can mislead scoring.
- **Coarse→fine search:** seed fine-scale alignment from coarse results; use **local refine windows** for speed and accuracy.



Baseline pyramid alignment (coarse→fine displacement refinement).



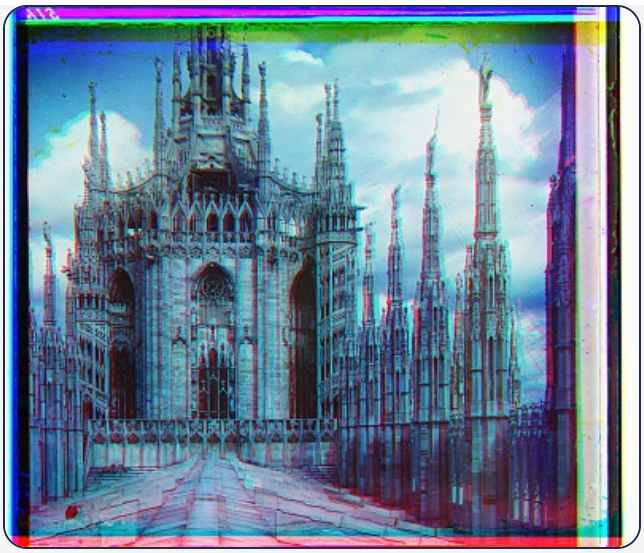
With edge features and careful slicing, refinement is crisper and less noisy.

Gallery

All aligned outputs with `_bells.jpg`. Captions show (green dy,dx) and (red dy,dx).

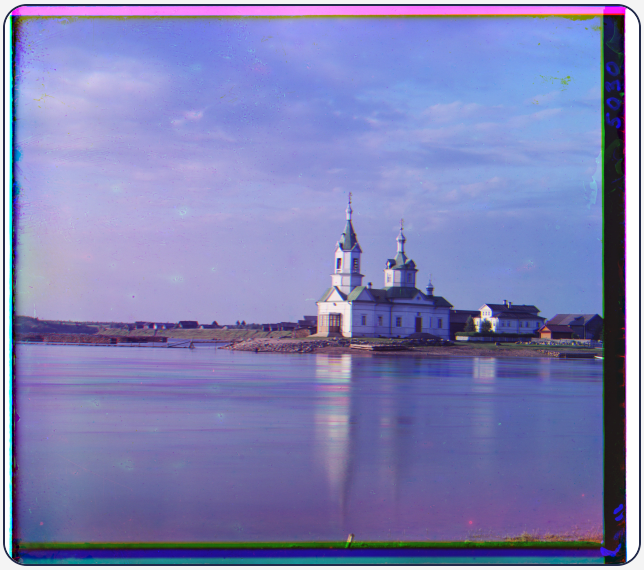
My Choice: 4 people — green offset: (5, 0) red offset: (8, -5)

My Choice: Khan — green offset: (1, 0) red offset: (1, 0). Note: this was the only image that the bells and whistles + image pyramid algorithm did not align. Naive brute force method worked much better, probably because the RGB negatives already have uniform brightness/contrast. Edge maps, cropping, and pyramids can add too much noise.



My Choice: Castle — green offset: (-1, -2) red offset: (0, -3)

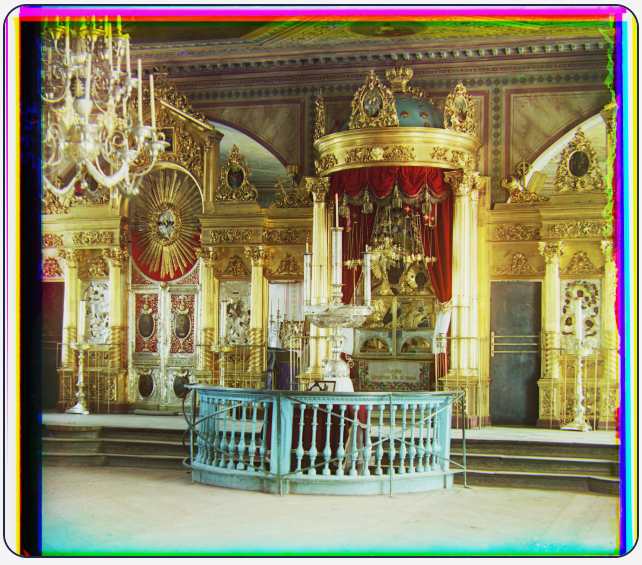
Cathedral — green (5, 2), red (12, 3)



Church — green (25, -2), red (58, -14)

Emir — green (49, 23), red (107, 40)

Harvesters — green (70, -6), red (123, 4)



Icon — green (42, 17), red (90, 23)

Italil — green (39, 22), red (77, 35)

Lastochikino — green (-3, -3), red (76, -8)

Lugano — green (41, -11), red (92, -29)

Melons — green (78, 5), red (177, 11)

Monastery — green (-3, 0), red (3, 0)

Self Portrait — green (78, -3), red (178, -1)

Siren — green (48, -8), red (96, -24)

Three Generations — green (54, 0), red (111, 7)

Tobolsk — green (3, 2), red (6, 3)

Fun proj, thanks course staff!