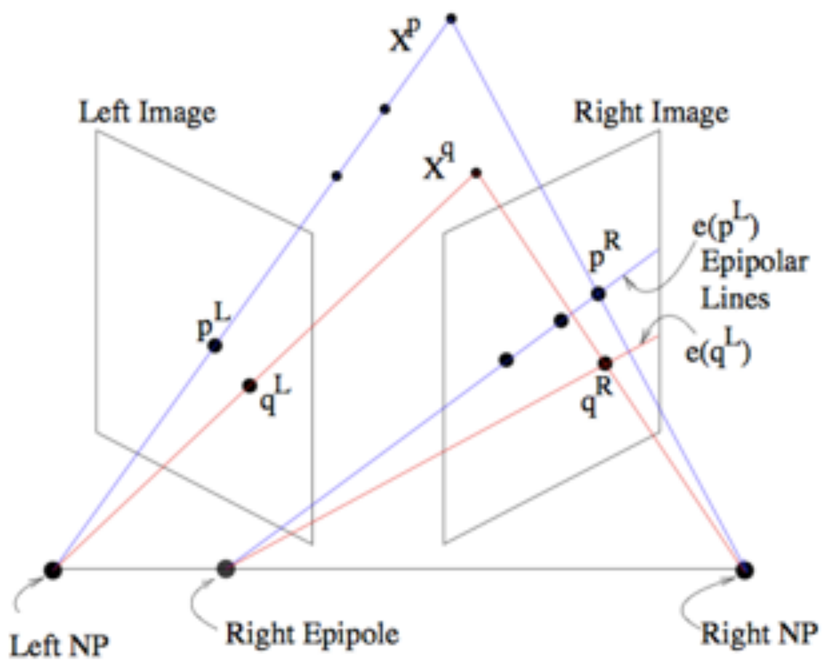
Projection, stereo images

X^p and X^q are the physical location of objects. **NP are the nadir points of the cameras.**



http://www.cs.toronto.edu/~jepson/csc420/notes/epiPolarGeom.pdf

The projection of the left nadir, NP is seen as a the right epipole with w.r.t. the right image. The epipoles may or may not be within the border of the images.

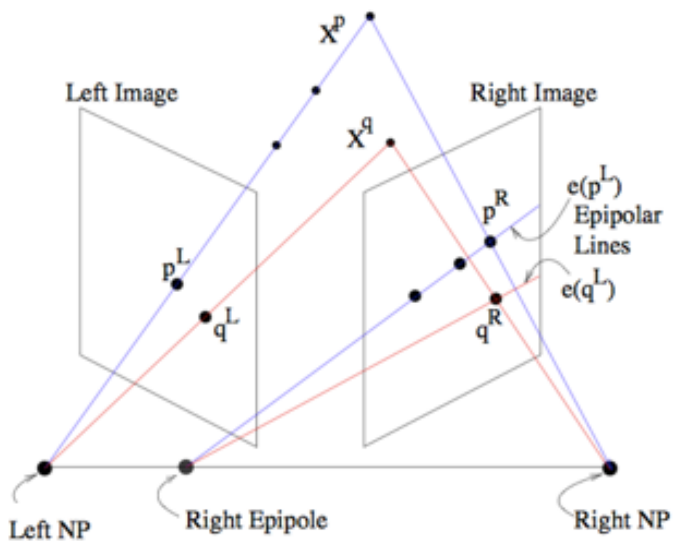
The projection of X^p to left nadir, NP is seen as an epipole line in the right image. If more than one epipole line is present in the right image, they converge at the right epipole (which might not be within the boundaries of the image).

Once the points in the left image, X_L are matched with points in the right image, X_R, if there are at least 7 points, one can determine the "bifocal tensor",
a.k.a. "fundamental matrix, relating the points in the 2 images using a 3x3 matrix of rank 2.

$(X_L)^T * F * X_R = 0$ for any pair or points in the images.

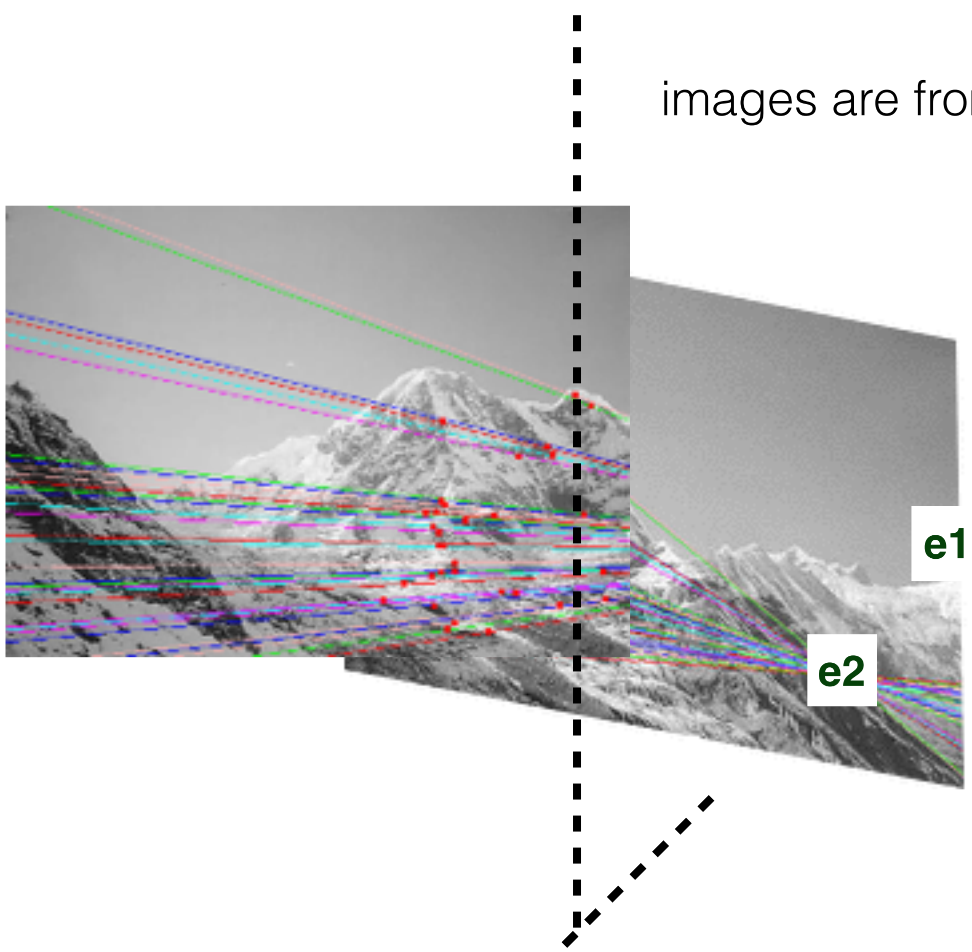
Note: the "Essential matrix" is a matrix used if the camera details are known. The "bifocal tensor", a.k.a. "fundamental matrix" does not need camera details.

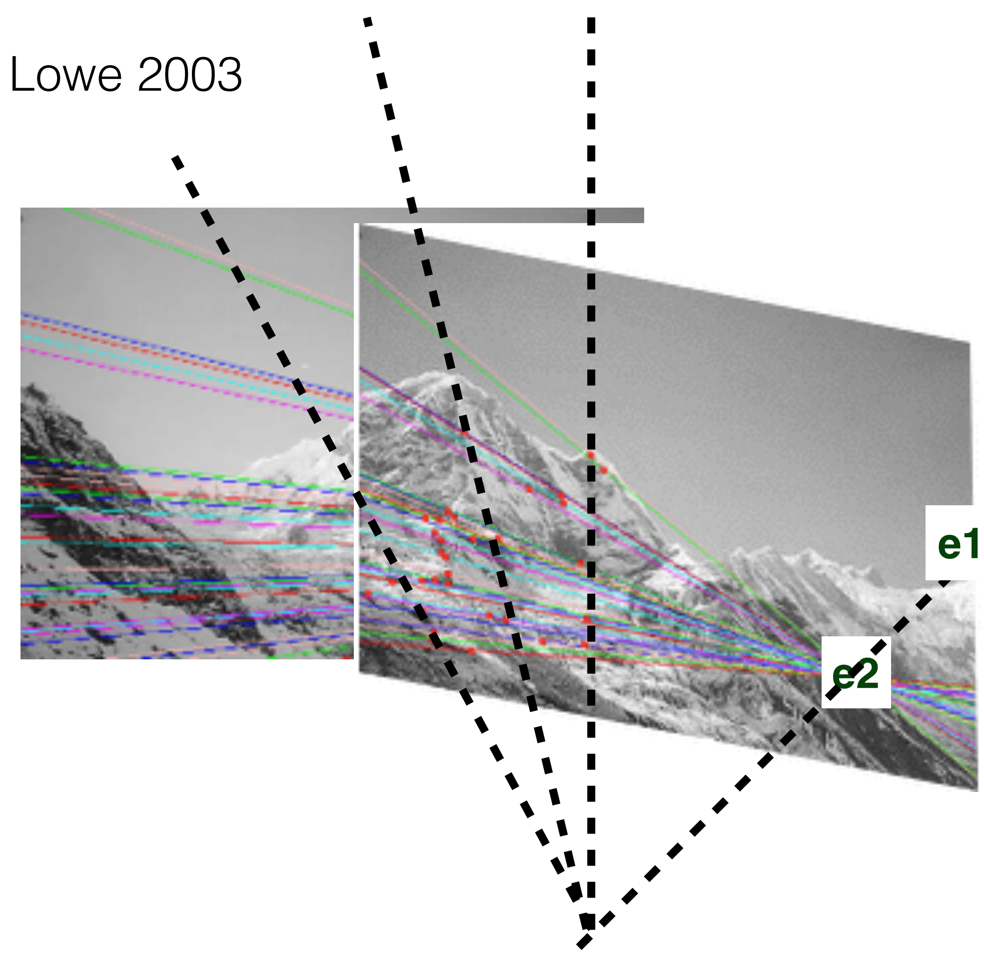
Projection, stereo images



http://www.cs.toronto.edu/~jepson/csc420/notes/epiPolarGeom.pdf

images are from Brown & Lowe 2003



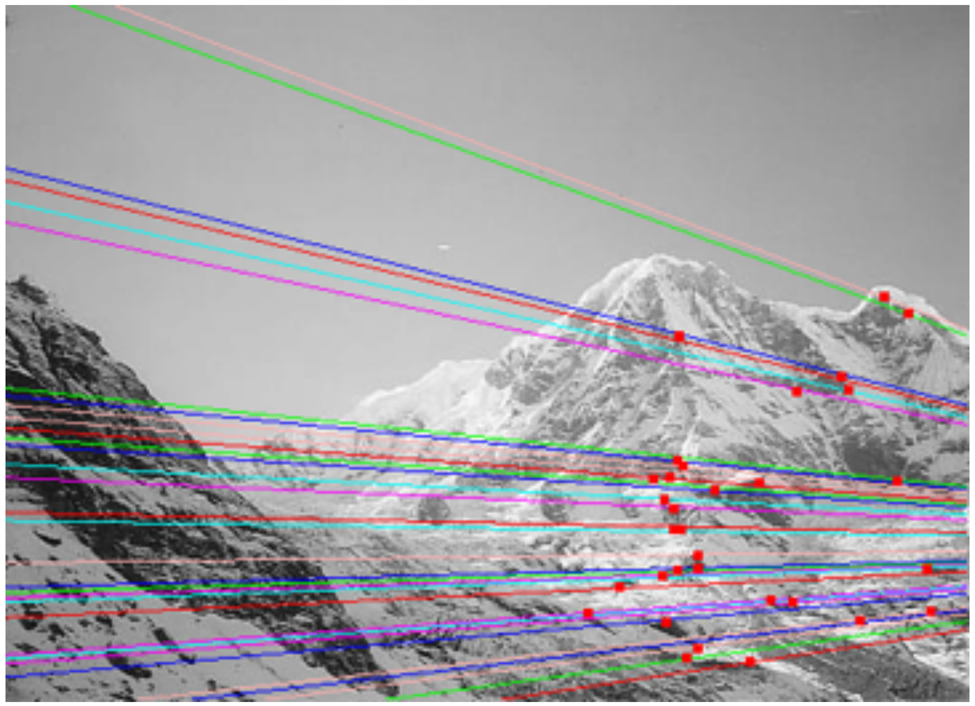


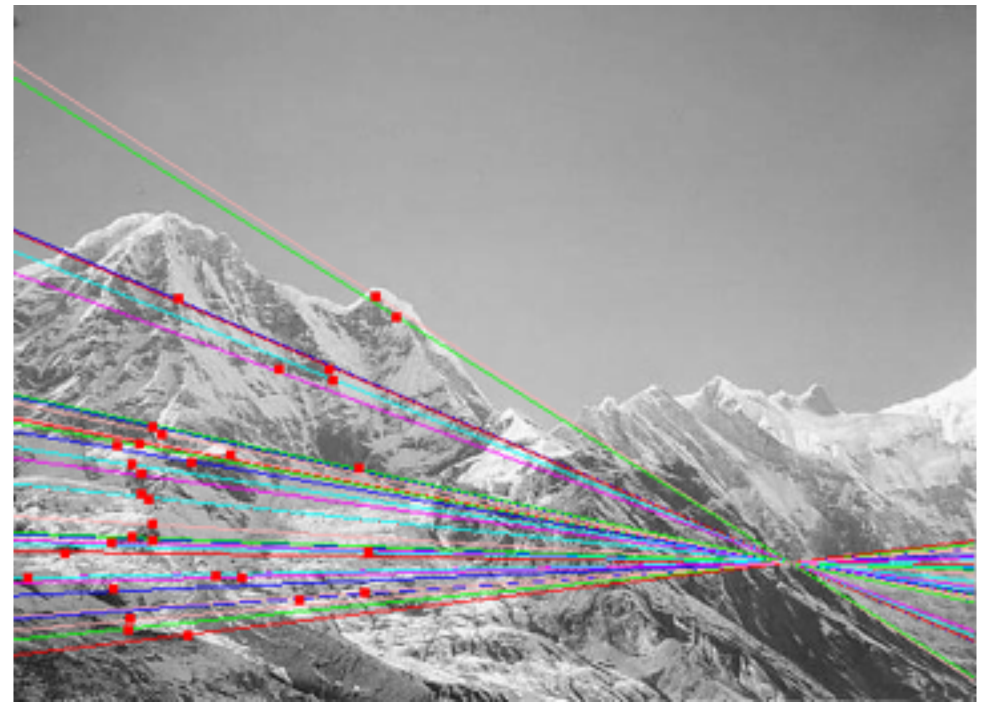
same camera objectives (nadirs), but different orientation for the 2 images (that is, rotated around the same nadir)

```
leftEpipole = (819.7904805976898, 289.2491982985311) = e1
rightEpipole = (-512.6627398191632, 11.894186211572105) = e2
```

Projection, stereo images

manually matching a subset of the corners from the edge extractor used with “outdoor mode”:

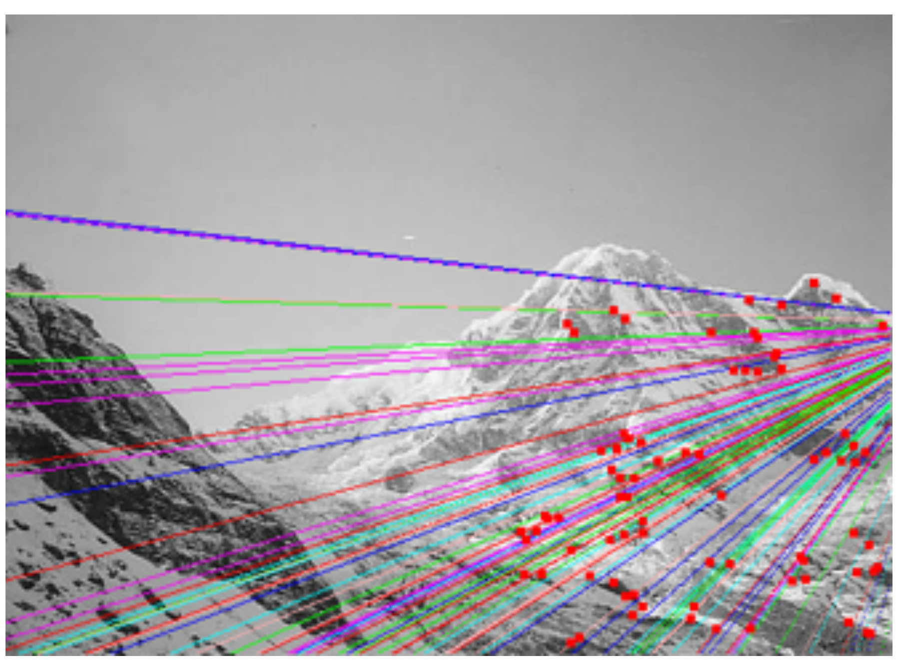


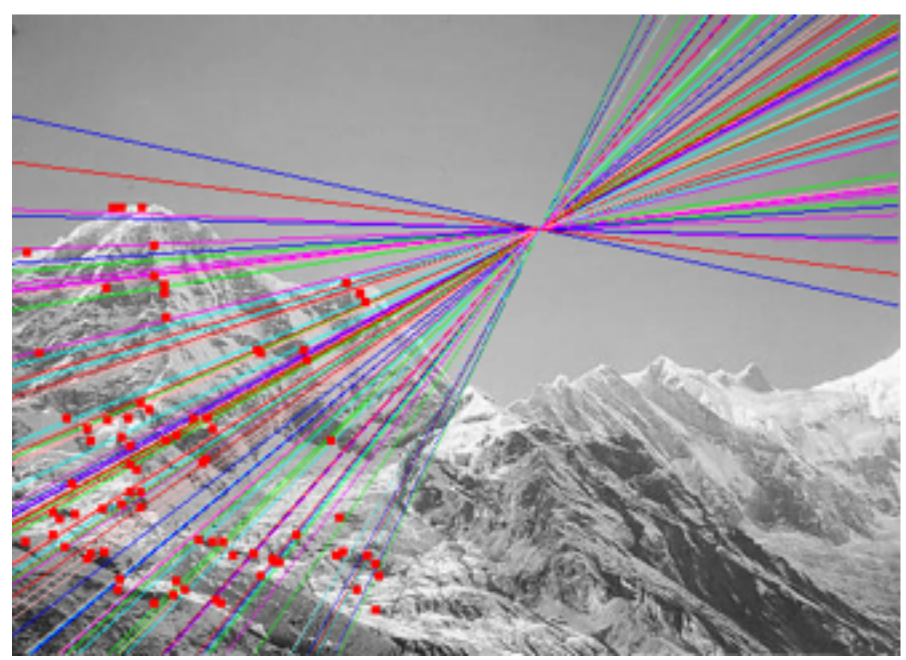


stereo projection fit to 32 points already known to match shows what the epipolar projections should be when the corner find + corner match + stereo projection solve are correctly automated.

```
nMatched=32
avgDist=0.281
stDev=0.508
```

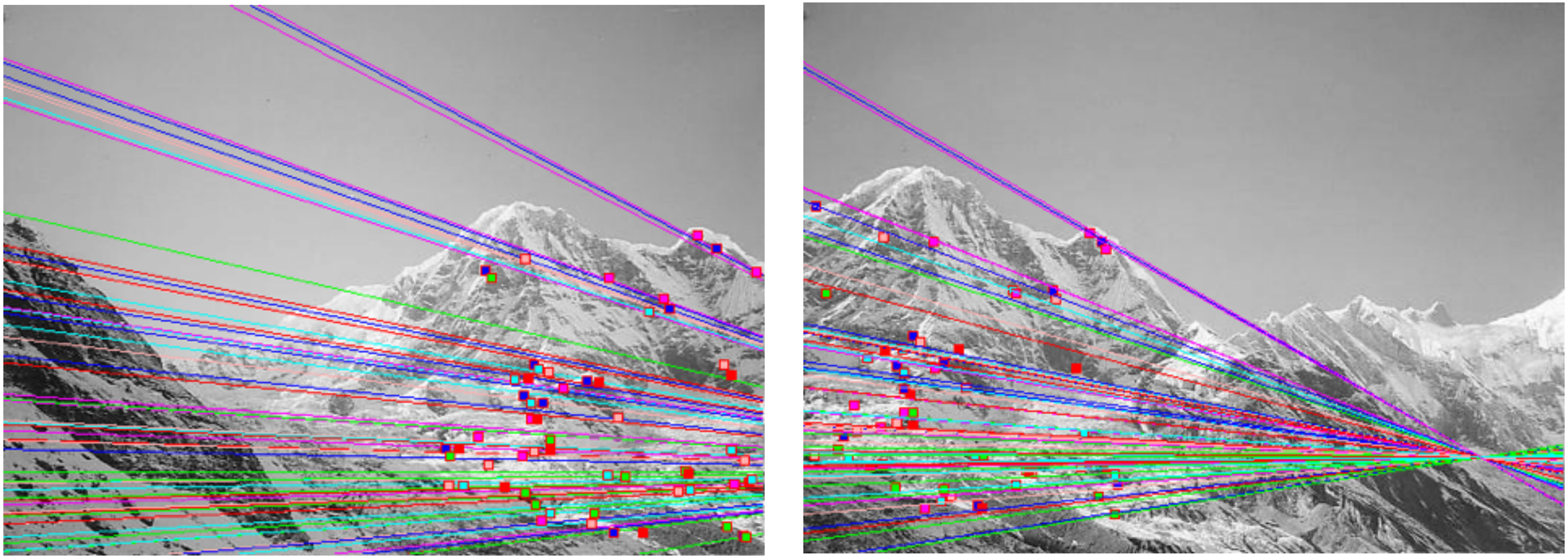
all corners from the edge extractor used with “outdoor mode”:





best of stereo projection iterations of pairings in the 78 un-matched points as a matrix, shows one cannot follow this w/ outlier removal from these epipolar line and point differences because the top points from the 2nd tallest mountain would be removed and those are 2 of the most important in an accurate fit (see above). So will use euclidean point matching first (same for inflection points).

Projection, stereo images



points of interest: **corners** calculated using extremes of curvature from edges created with "outdoor mode".

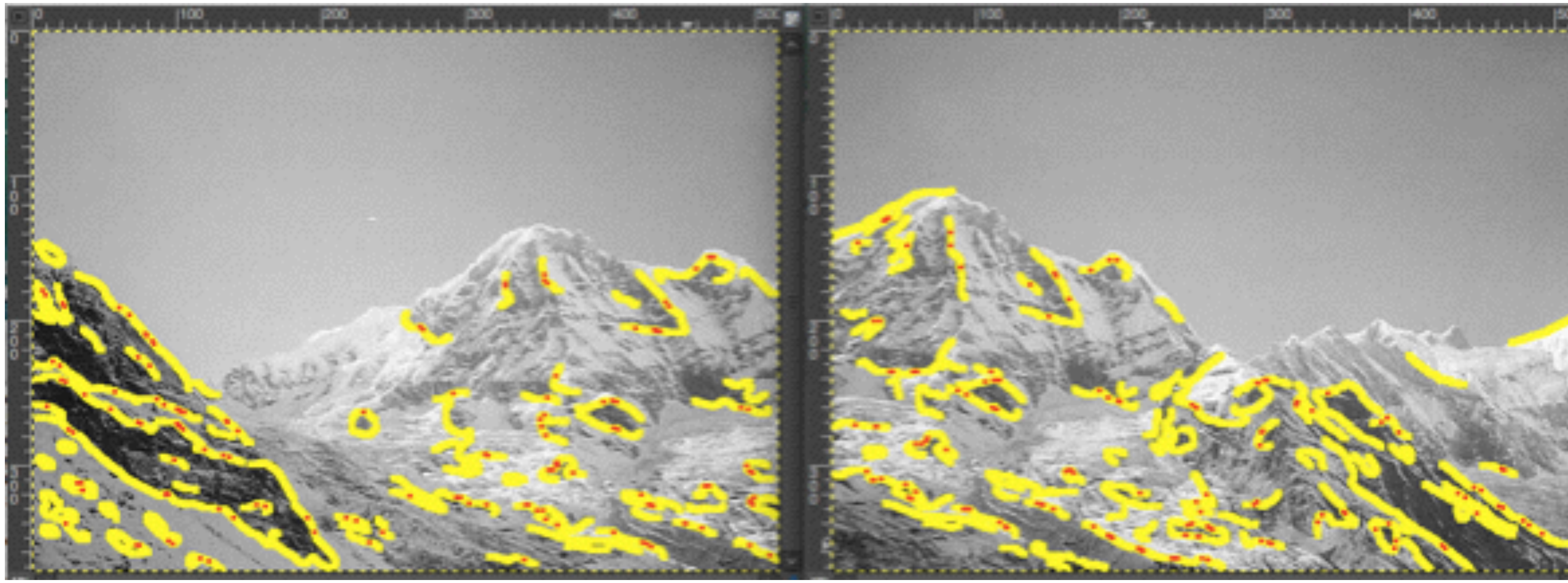
point matcher: find rough euclidean transformation, then rough match of sets, then improved euclidean transformation, followed with removal of points with residuals > avg dist + 0.5*standard deviation.

stereo projection: the above is one invocation of the stereo projection calculation (no further outlier removal).

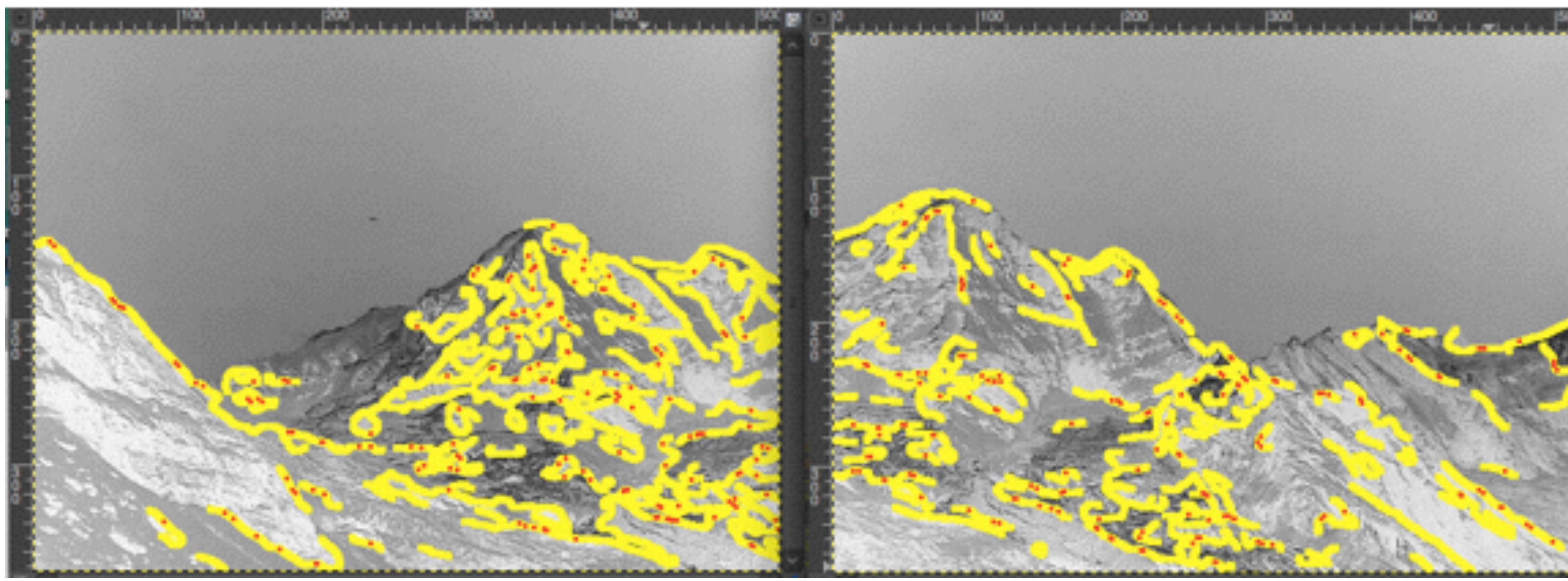
```
nMatchedPoints=60
meanDistFromModel=5.683 (compared to the manually chosen result 0.281)
stDevFromMean=6.138 (compared to the manually chosen result 0.508)
```

corner sets have few common members (even from the specially tailored "outdoor mode" edges) so the sets are difficult to match. Trying inflection points next.

points of interest:
inflection points calculated using the peaks from the contours extracted from scale space images. The edges used to create the scale space images were created with "outdoor mode".

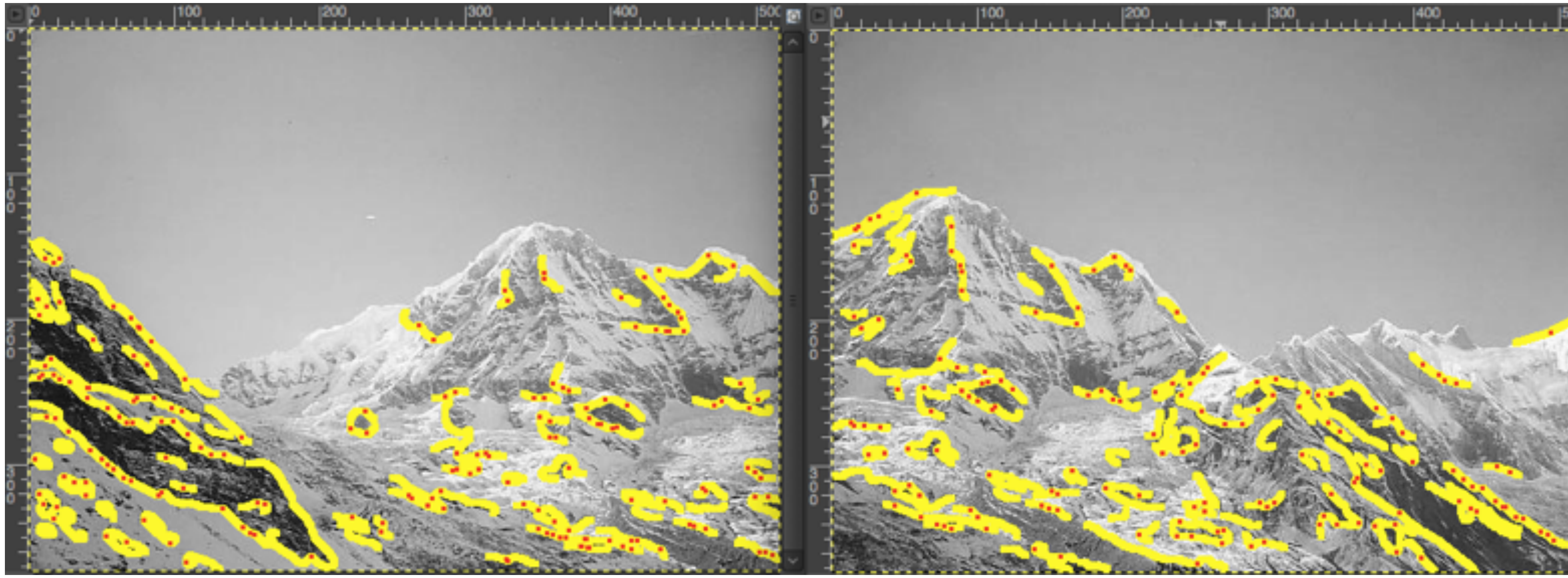


inflection points created from the *inverse* images

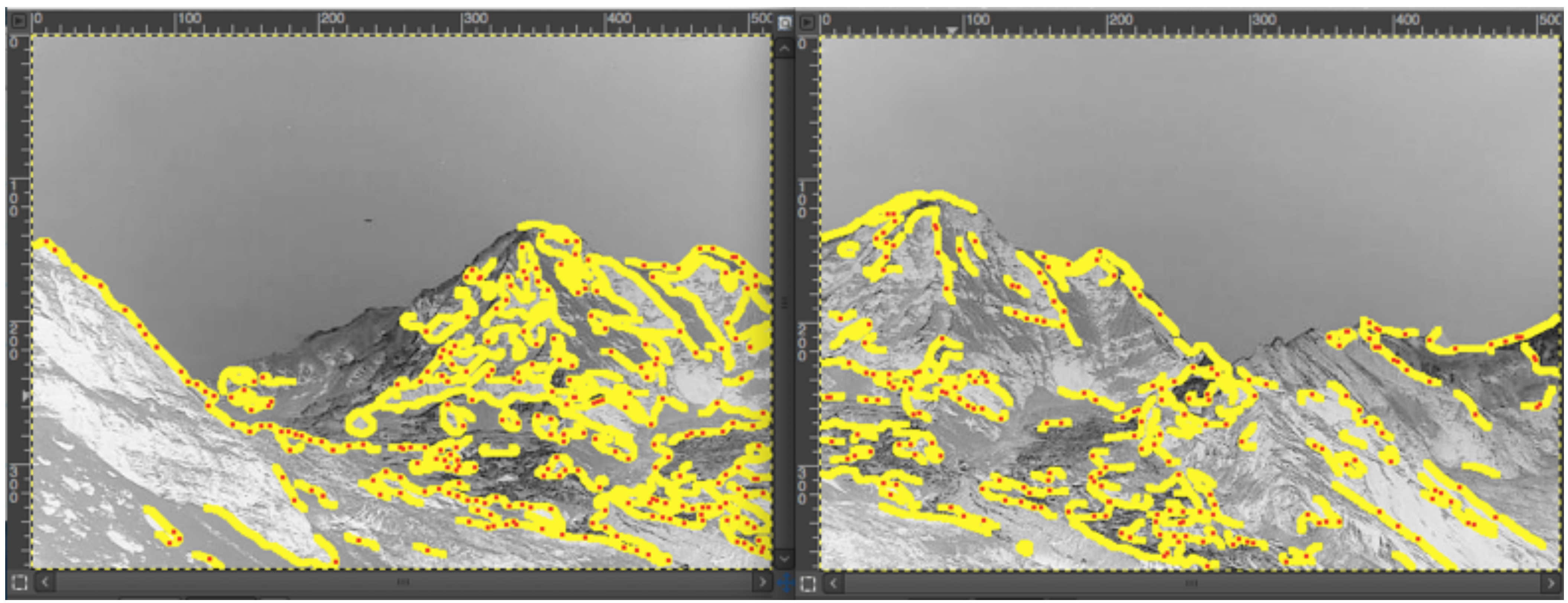


inflection points are more fragile to intensity levels (for higher density of edges, there are complex junctions in some and for those, edges may not connect, which prevents finding inflection points). See the difference in left and right points for the inverse of the image (bottom panel). It would be difficult to match the inverse image points, which suggests this would not be a robust feature maker.

points of interest: corners
The edges were created with "outdoor mode".



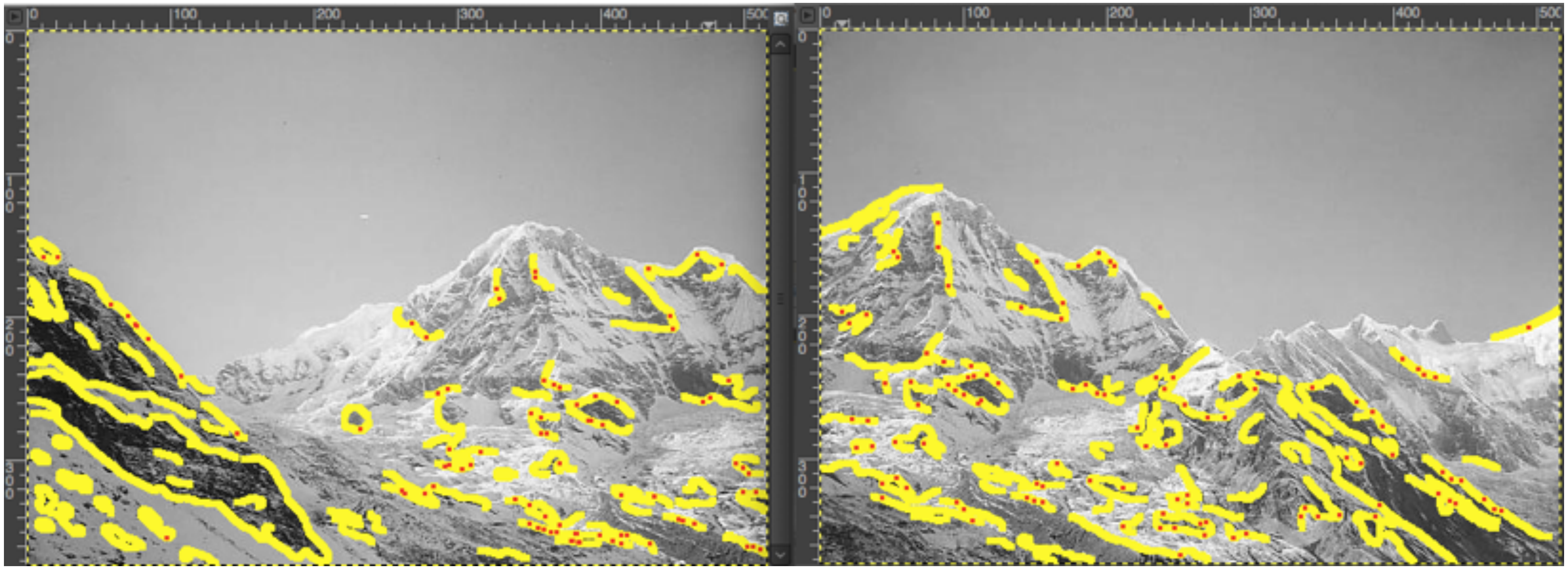
corners
created from the *inverse* images



corners appear to be more robust to high edge densities than inflection points. The difference in inverse image corners and the image corners suggest that the corners in common to both might be the best features to use for matching. (see next page).

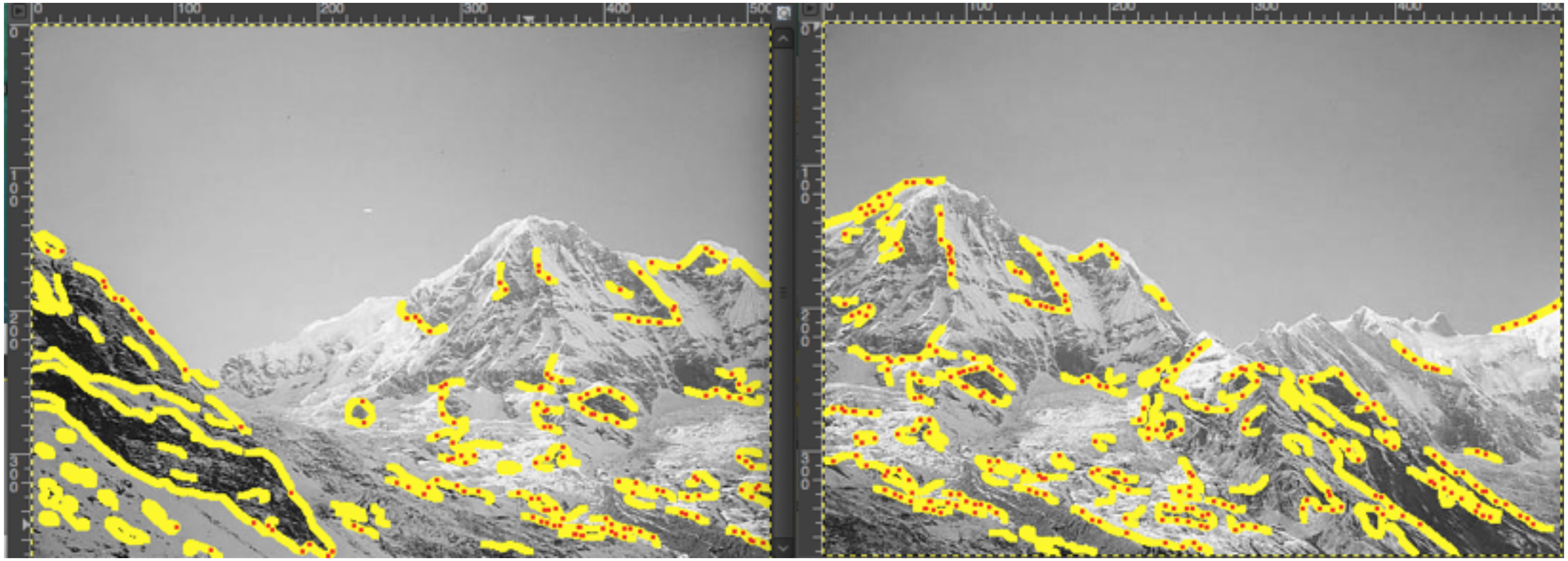
points of interest: corners
The edges were created with “outdoor mode”.

Corners were created for the <u>images and</u> then the <u>*inverse* of the images</u> and only the corners which appear in both are kept (tolerance is +- 10 pixels).



This method of image corners and inverse image corners intersection looks promising. caveat is it takes twice the amount of time to create the corners.

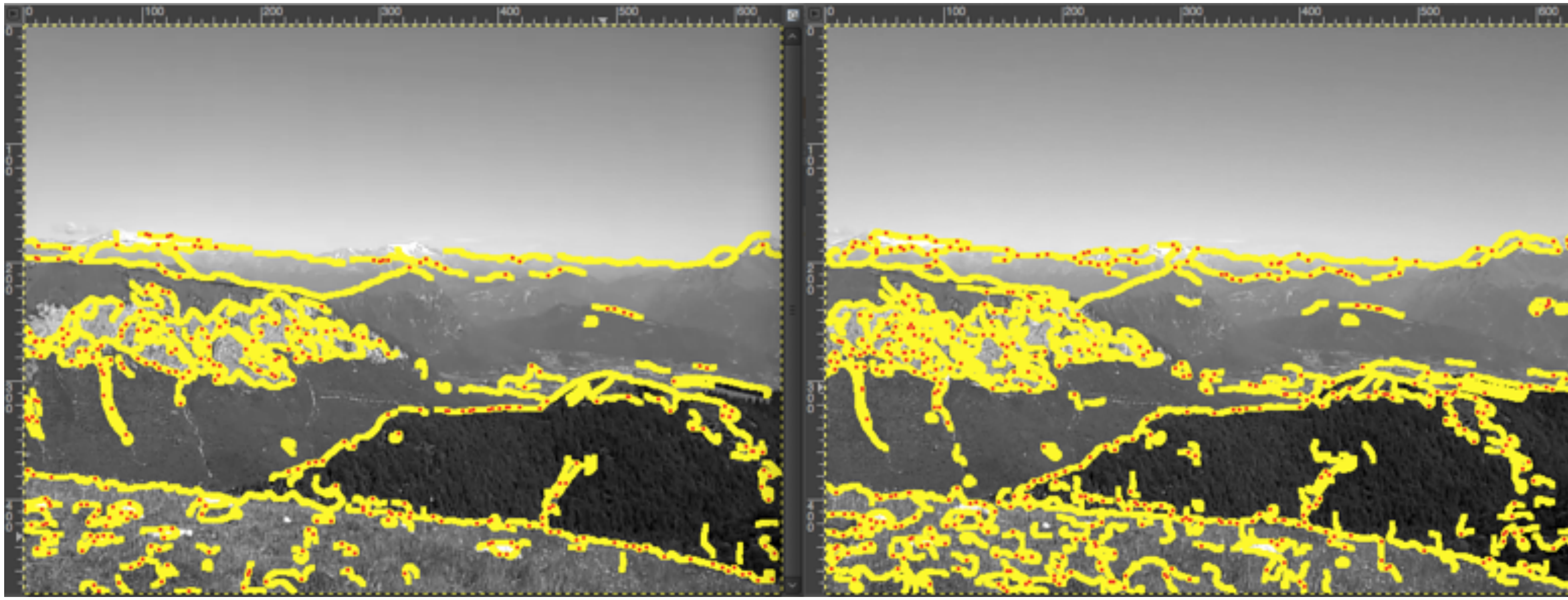
points of interest: corners
The edges were created with “outdoor mode”, but using normal curvature threshold and no inverse image corners.



points of interest: corners

The edges were created with "outdoor mode".

Corners were created for the images and then the *inverse* of the images and only the corners which appear in both are kept (tolerance is +- 10 pixels).



This is a difficult image set with many foreground and background details. The resulting points do look like they would lead to a match, mostly due to the patterns formed by neighboring points such as the ridge in lower middle of image.

http://www.cex3d.net/inverse/technical/cameraregistrationfromimagecorrespondences_report.pdf

6.7.1 The RANSAC Algorithm

RANSAC works by randomly taking out a sample set S_i of point correspondences, estimating a fundamental matrix F_i from those and checking how many of all the point correspondences are consistent with the estimated matrix F_i. This process is iterated N times, in search of the estimated fundamental matrix F_*, which were consistent with the largest number of point correspondences. When using the 7-point algorithm described in section 6.4.1, the size s of the sample set needs to be only seven point correspondences. The number N of iterations may be determined adaptively. The criterion for choosing N should ensure a very high chance of having found the fundamental matrix with the largest support,

i.e. being consistent with the largest number of point correspondences. The algorithm is as follows (sections 4.7.1 p. 117-121 and 11.6 p. 290-291 in [Hart03]):

- Initially set $N = \infty$ (total number of iterations) and $i = 0$ (completed iterations)
- Iterate the following, while $N > i$:
  - Randomly select a sample set S_i of seven point correspondences. See the description below
  - Estimate the fundamental matrix F_i from S_i. When using the 7-point algorithm, up to three fundamental matrices F_{i_j} may be estimated. For each estimated F_{i_j} do the following:
    * Compute the distance pairs $(d1_{i_{j_k}}, d2_{i_{j_k}})$ of the putative correspondences $\mathbf{x}_k \leftrightarrow \mathbf{x}'_k$ from F_{i_j} and the paired reprojection error, explained in equation 8 in section 6.7.3
    * Determine the inliers, i.e. the point correspondences consistent with F_{i_j}. These are the correspondences where $d1_{i_{j_k}} < t_1$ and $d2_{i_{j_k}} < t_2$. Section 6.7.3 describes the distance thresholds t_1 and t_2. Using two thresholds here differs from algorithm 11.4 in [Hart03]
    * If the number of inliers is higher than any previous number of inliers, these inliers and the matrix F_{i_j} are now remembered as the best set of inliers and the best fundamental matrix $F_* = F_{i_j}$. In case of an equal number of inliers, the set with the lowest standard deviation of inlier distances is chosen. The standard deviation calculation is computed by equation 10 in section 6.7.3, where the two distances $d1_{i_{j_k}}$ and $d2_{i_{j_k}}$ from the distance pairs are added as $d_{i_{j_k}} = d1_{i_{j_k}} + d2_{i_{j_k}}$
  - Set $\epsilon = 1 -$(best number of inliers)/(total number of point correspondences)
  - Adaptively determine N from ϵ by the formula below in equation 7 with $p = 0.99$
  - Increment i by 1 for the next iteration

Images with projection still require point matching before an attempt to calculate the epipolar projection.

Point matching: when the image region differences are as large or larger than the intersection of the image regions, if there is projection, a simple euclidean solution using all points may fail. For that reason, partitioning the image into a few sections and solving the combinations of partitions is an improvement that does not add too much more to the runtime.

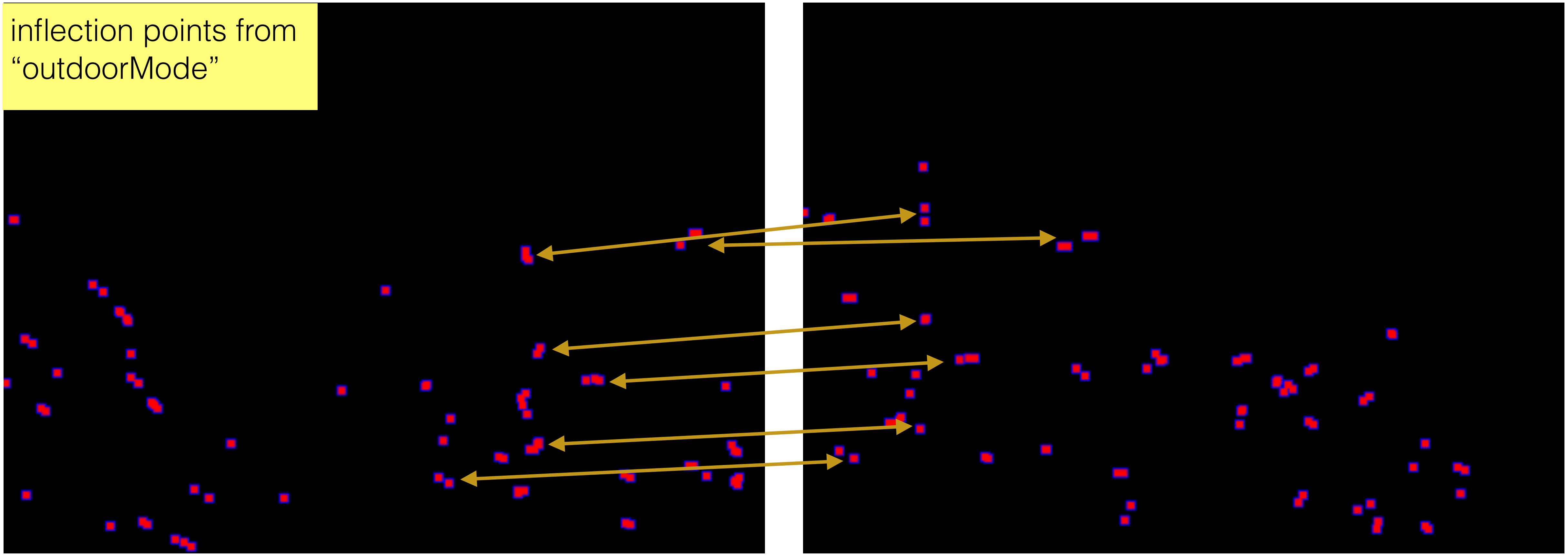
The number of partitions should be kept low, so the partitions formed will be
- (1) vertically halved for each image.
- (2) horizontally halved for each image.
- (3) quadrants for each image.

For each of the (1) thru (3) partition combinations:
- — finding the best fitting euclidean transformation w/ a cost function that uses the residuals between transformed and model instead of using the number of matched points.
- — the euclidean transformation and a large tolerance is used to make matched points lists.
- — RANSAC w/ a threshold of 5 is used with the matched point sets to estimate the projective transformation.
- — the sets of all points are filtered to the region of overlap for the image sets and the filtered points are used to evaluate the projective transformation.
    - — the statistic is number matched / max matchable in intersection region

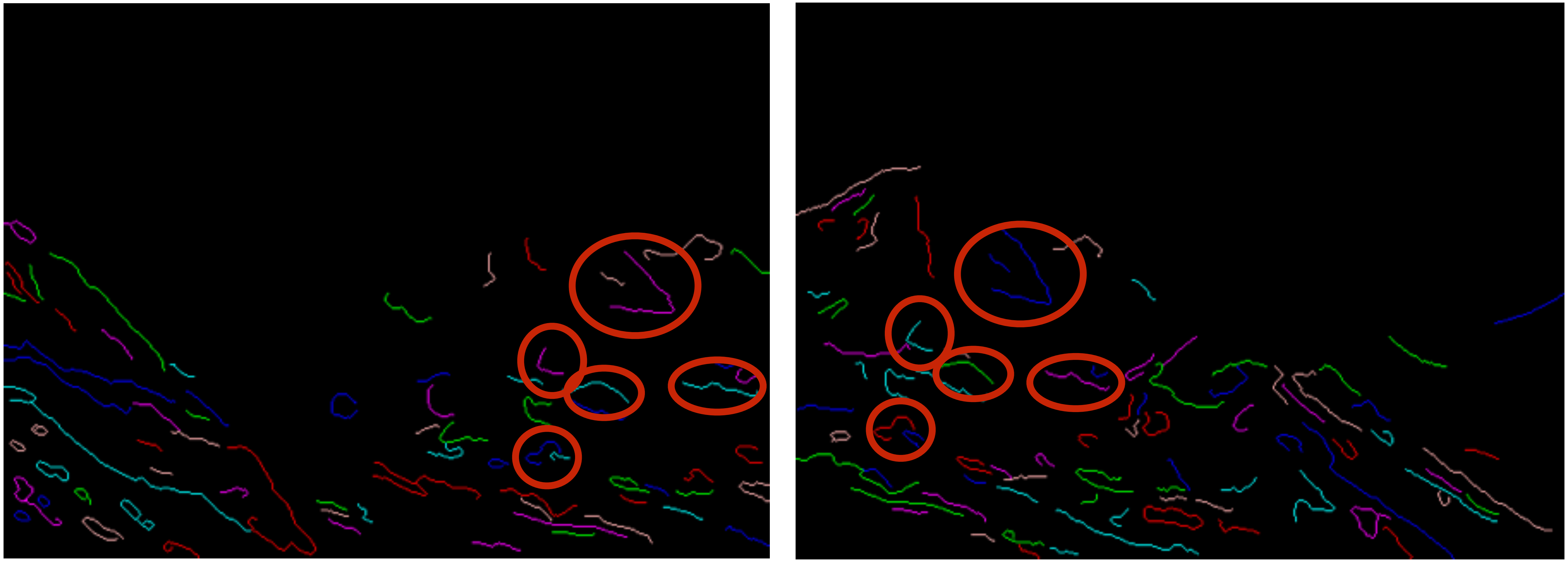
The best projective solution of all partitions is kept.

Not Finished… in progress

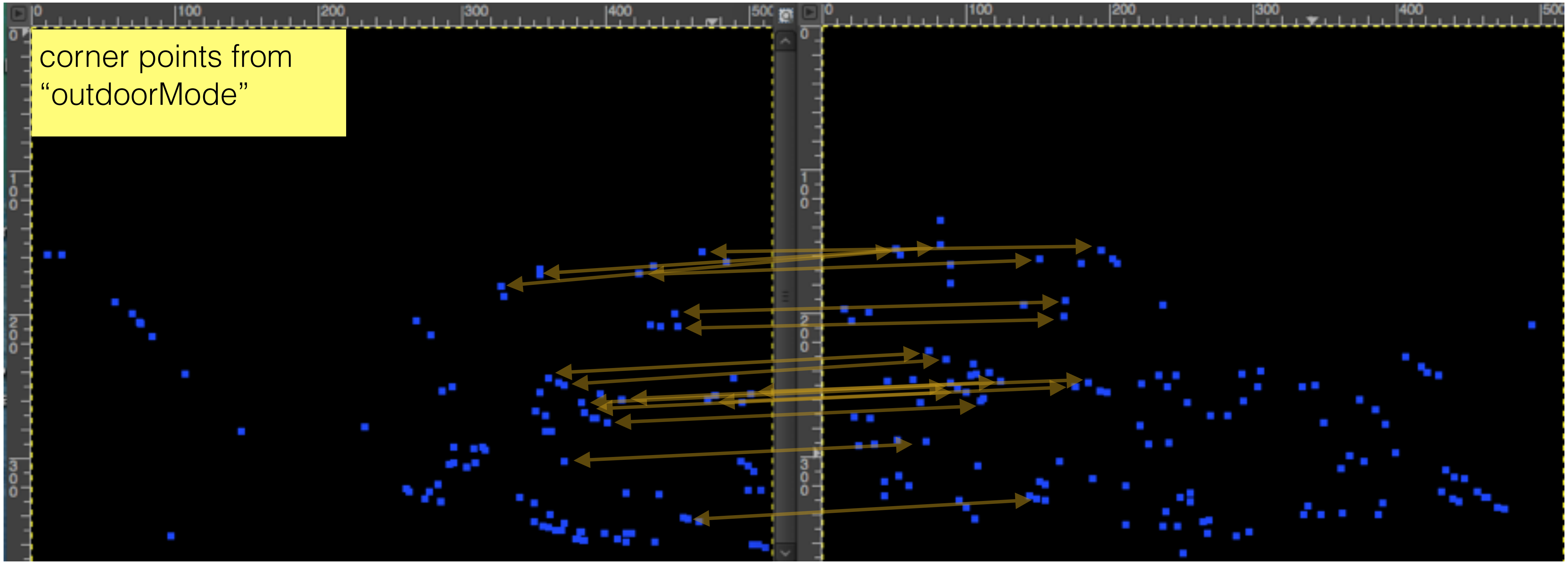
The Brown & Lowe 2003 images: point matching difficult because image intersection << difference



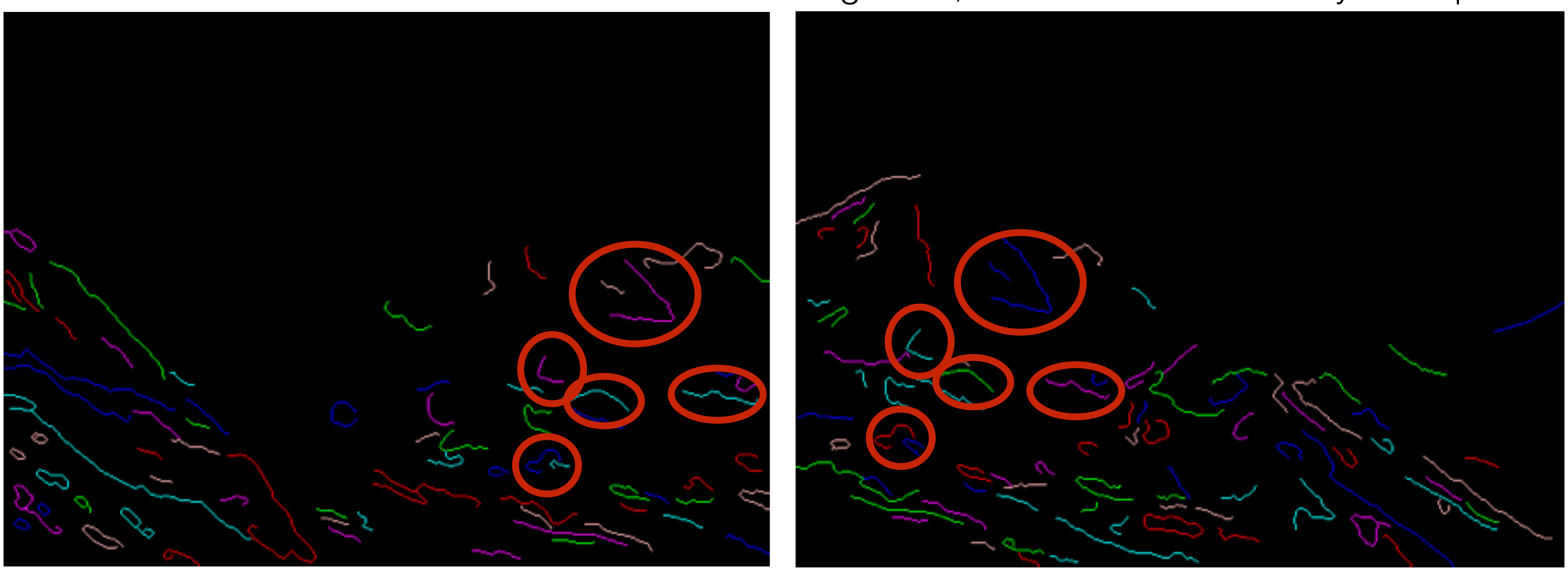
could consider distinct curves and their immediate neighbors, but that would be many more points:



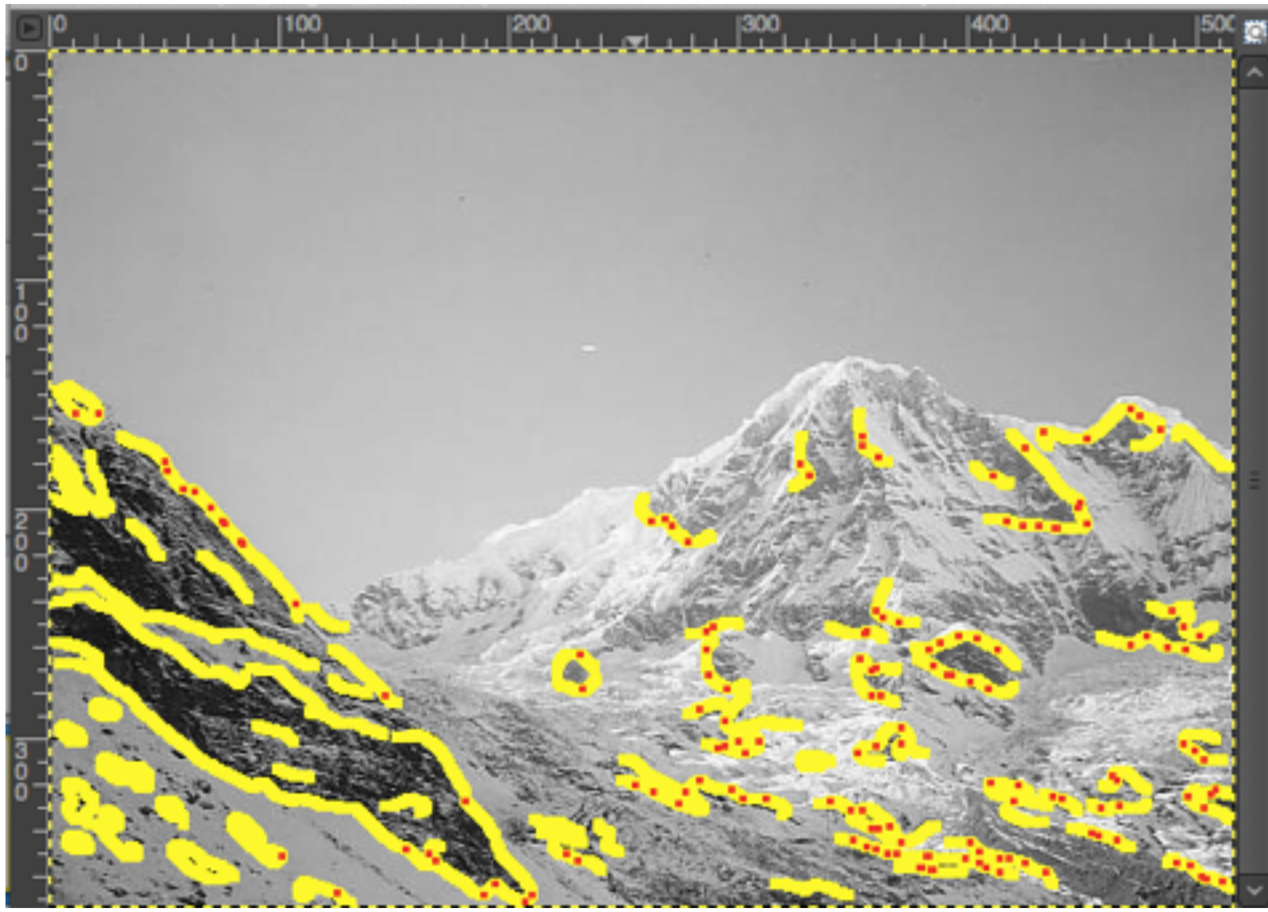
The Brown & Lowe 2003 images: point matching difficult because image intersection << difference

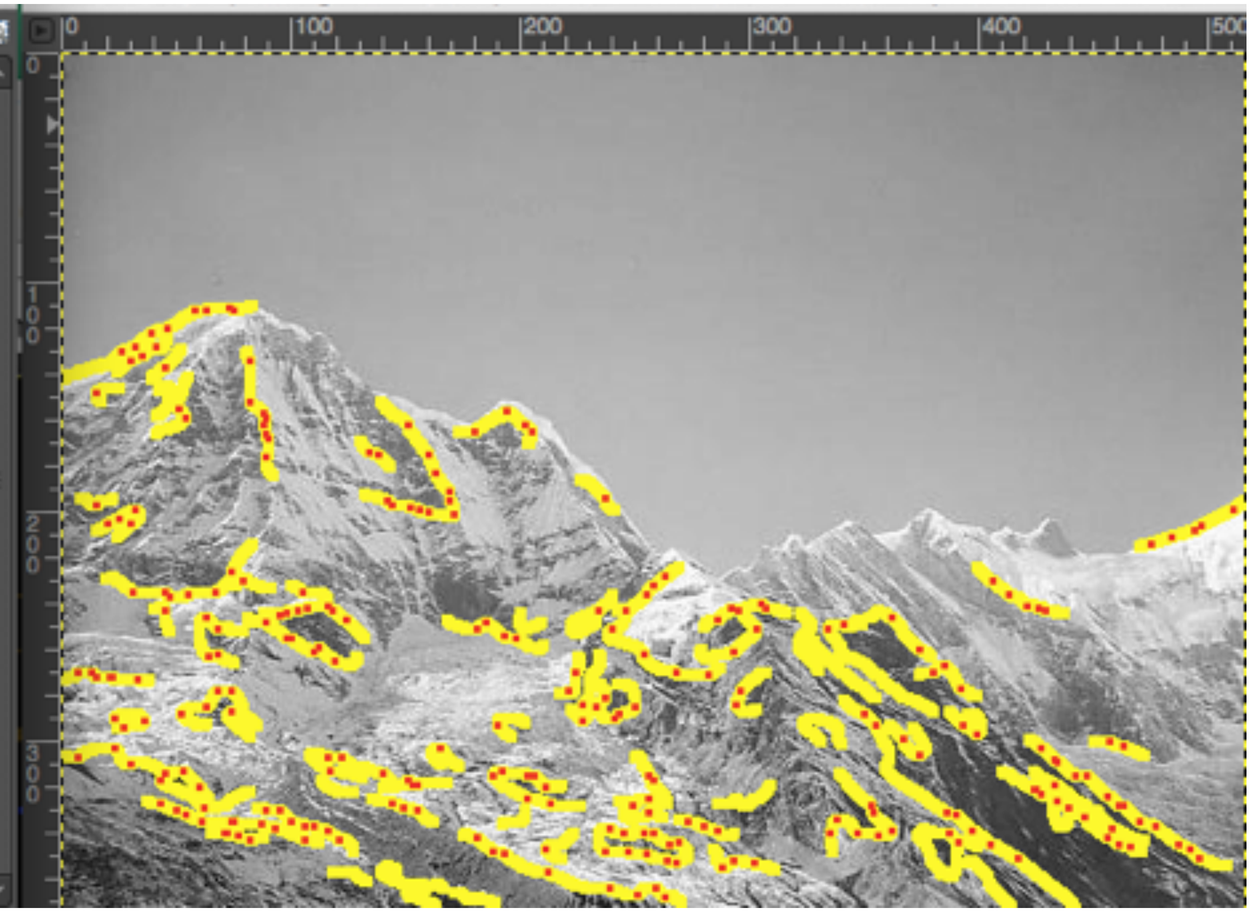


could consider distinct curves and their immediate neighbors, but that would be many more points:

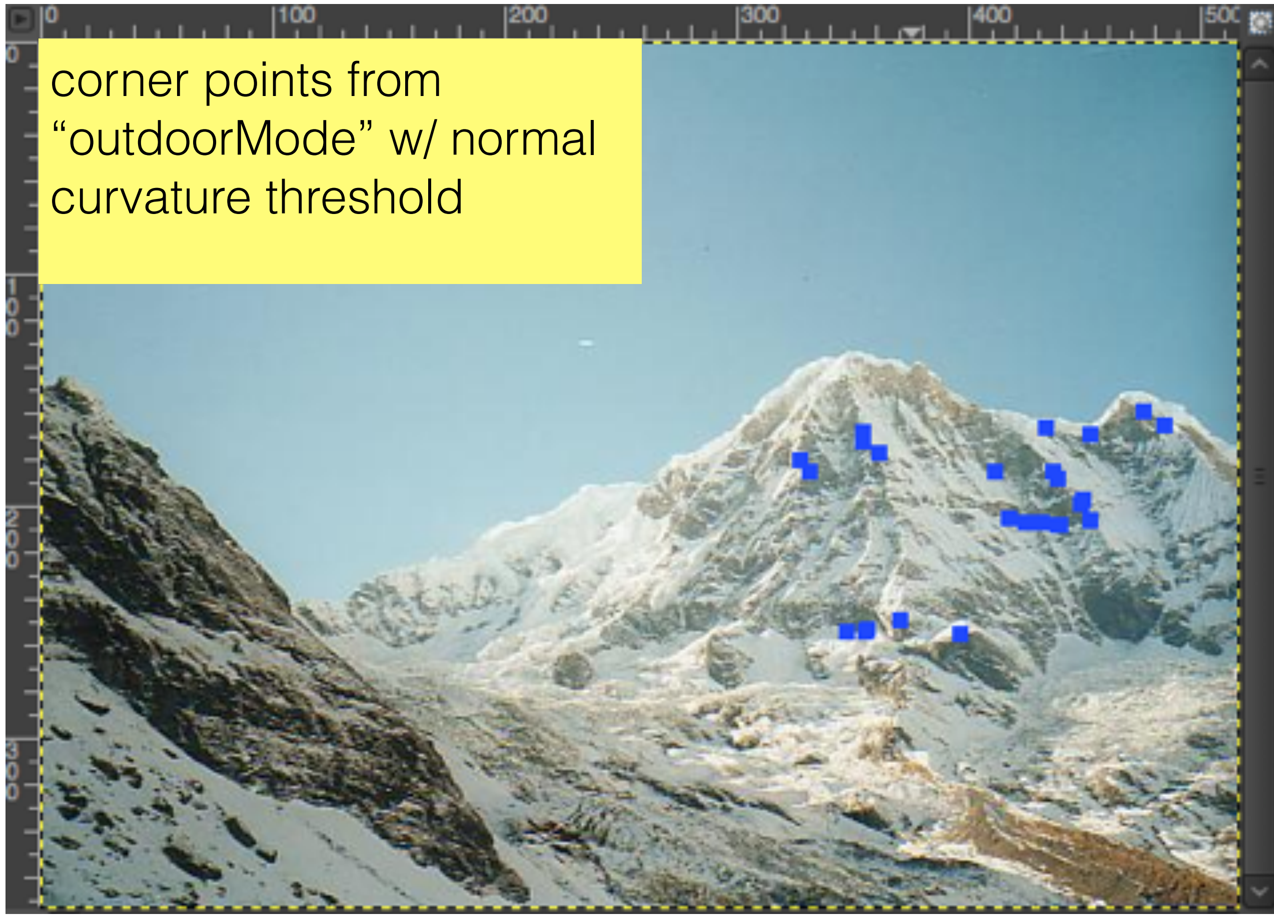


changes in corner detector for “outdoorMode” to include corners at a lower threshold (relative curvature strength). The threshold is now the same as for the corner detector in default mode.

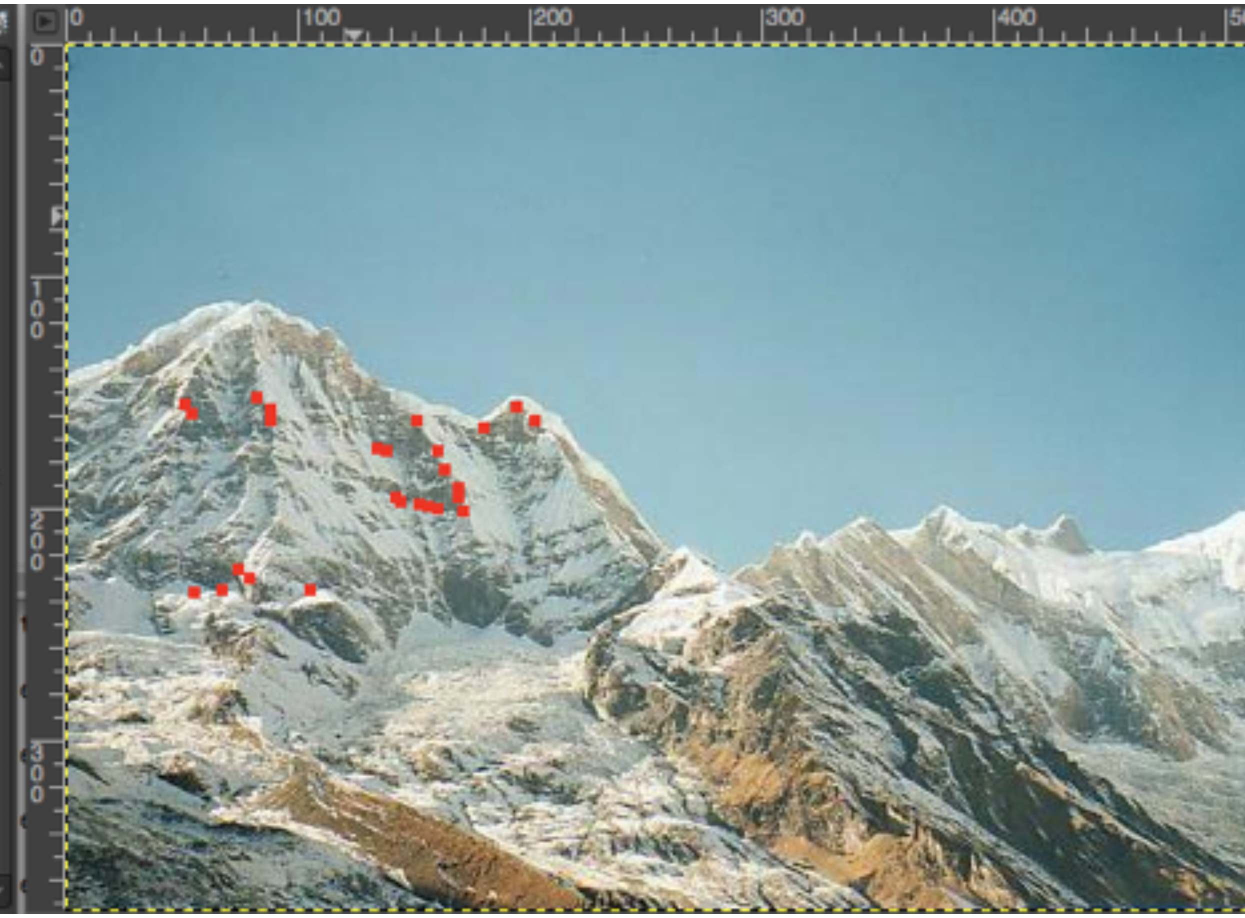




partitioning by quadrants, these 2 quads are the matches.

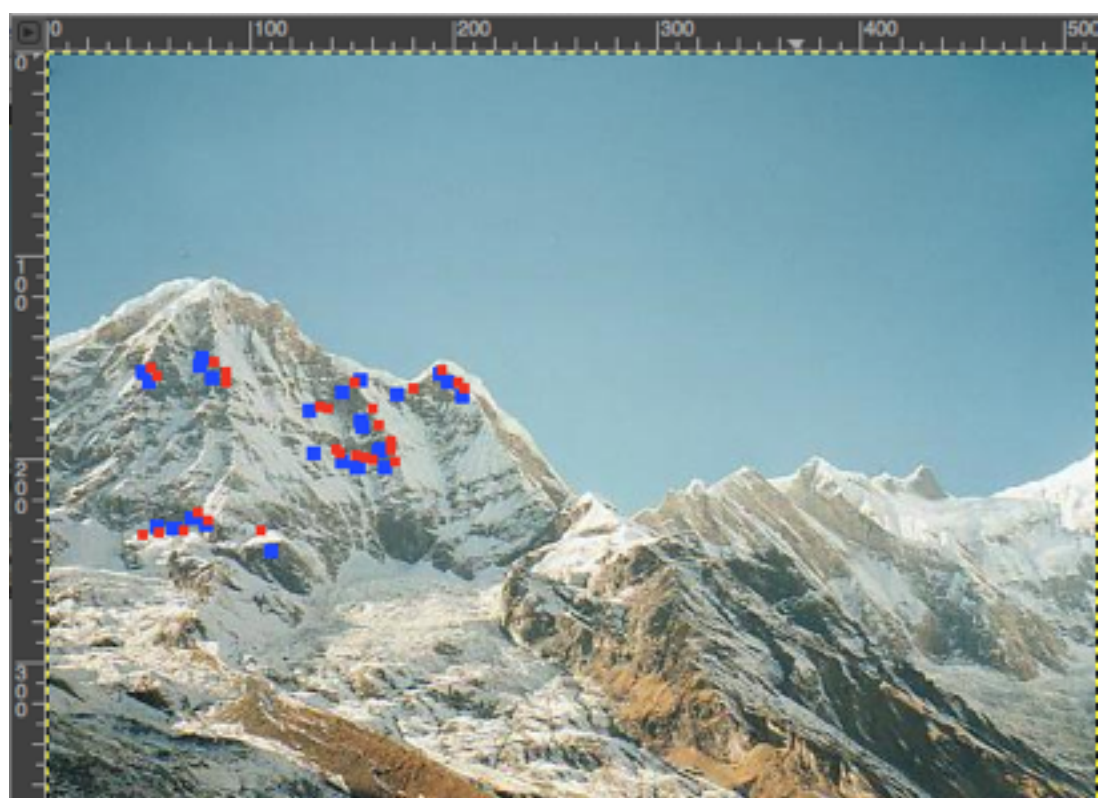


can see 2 false matches. RANSAC might discard them.



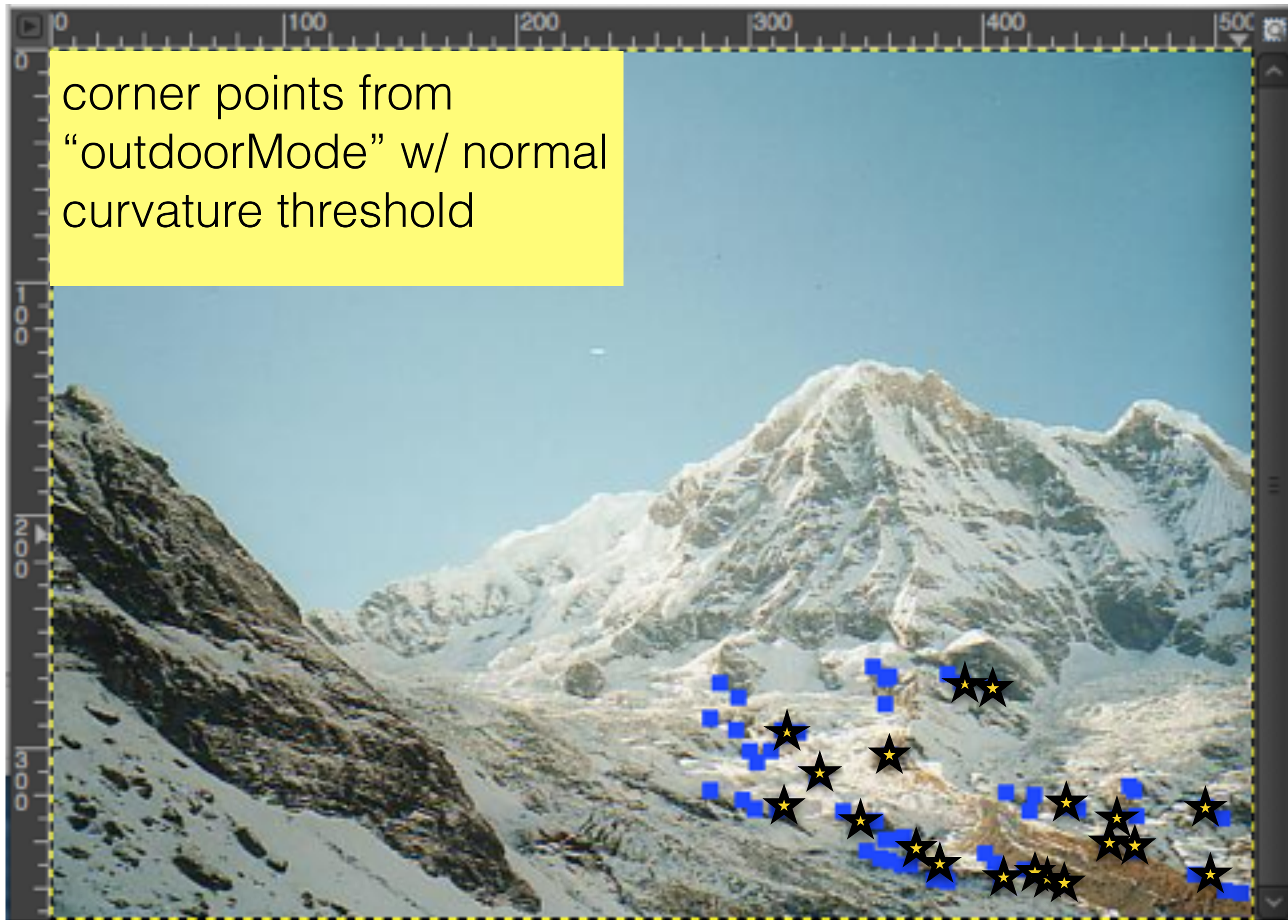
euclidean
scale=1,
rot=350,
transx=-280,
transy=+35.

optimal
matching
w/
toler = 15



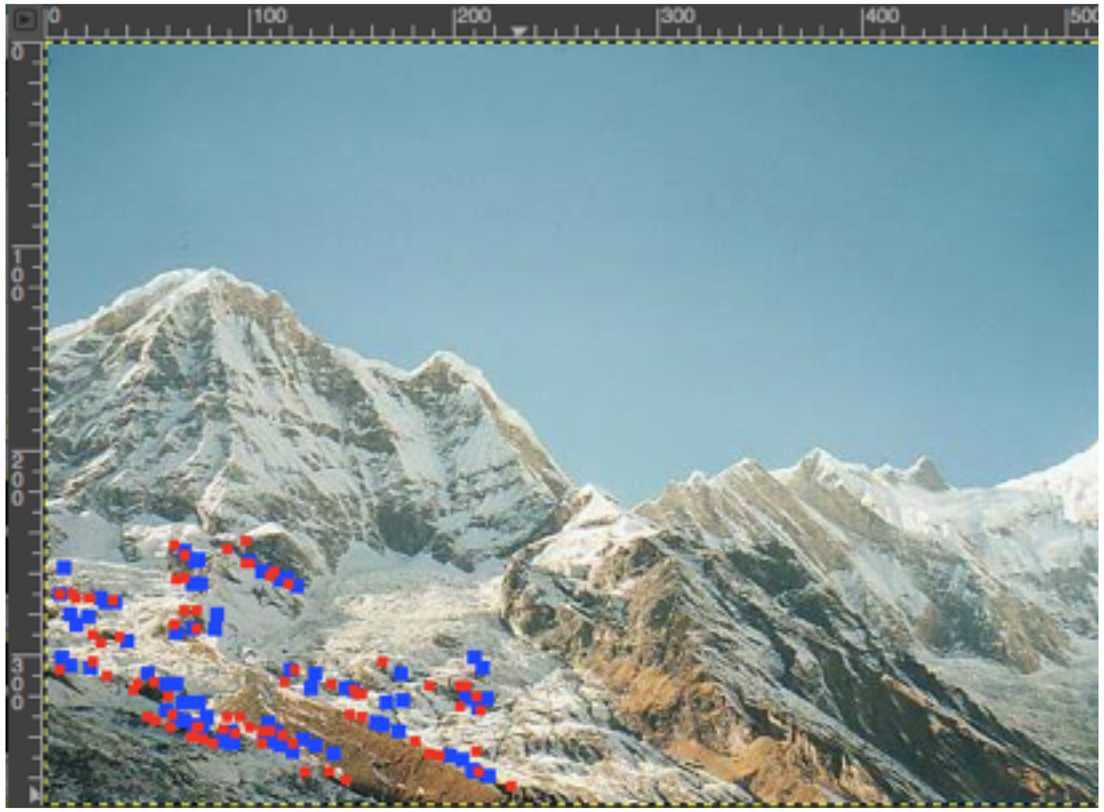
tolerance of
4 would give
better
results for
this section,
but the
bottom
matches on
next page
need larger
tolerance.

partitioning by quadrants, these 2 quads are the matches. can see some false matches. RANSAC might discard them.

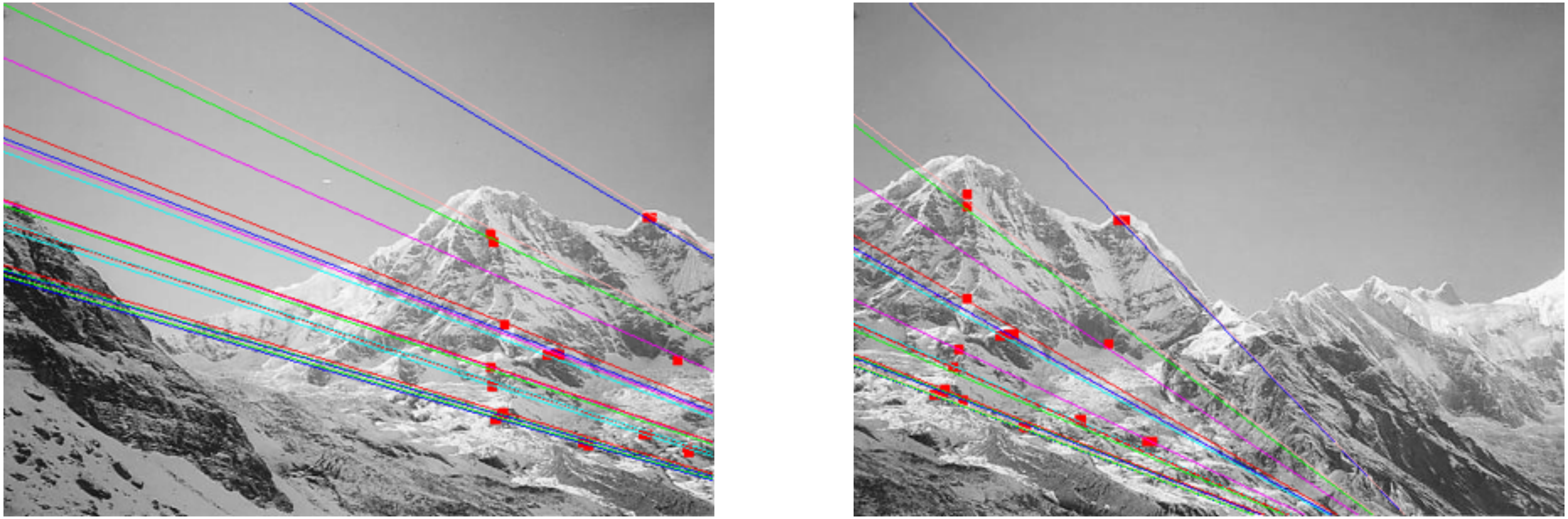


euclidean
scale=1,
rot=355,
transx=-277.4 (bottom needs -10 more)
transy=-24.3

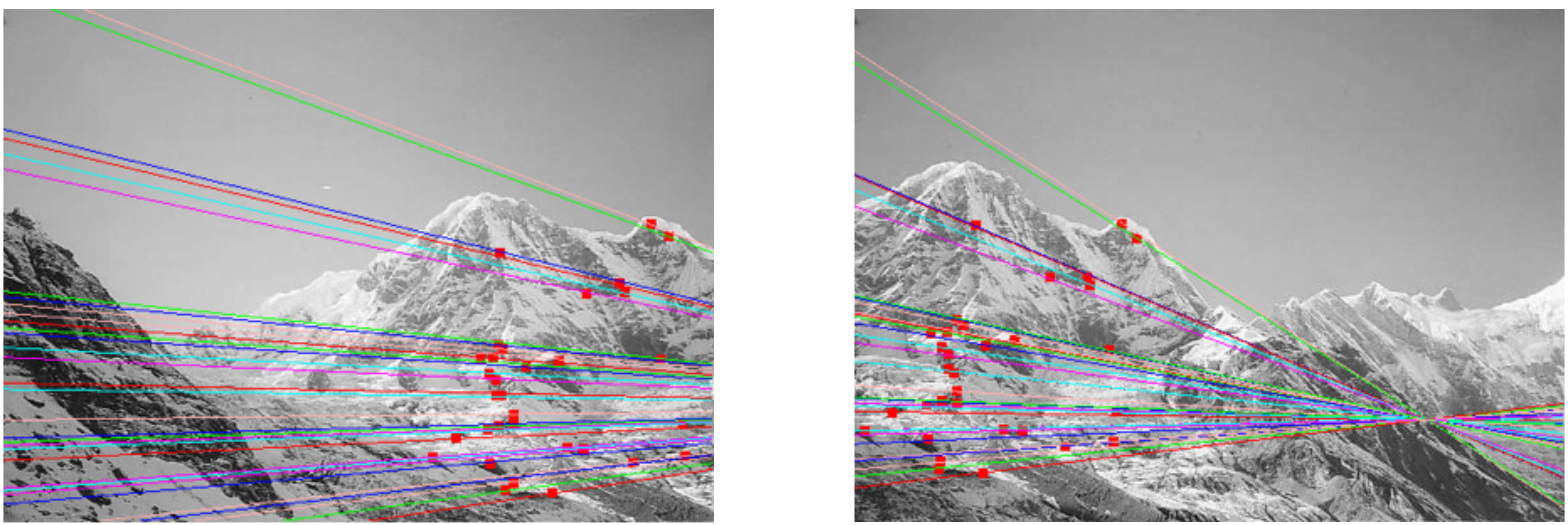
optimal
matching
w/
toler = 15



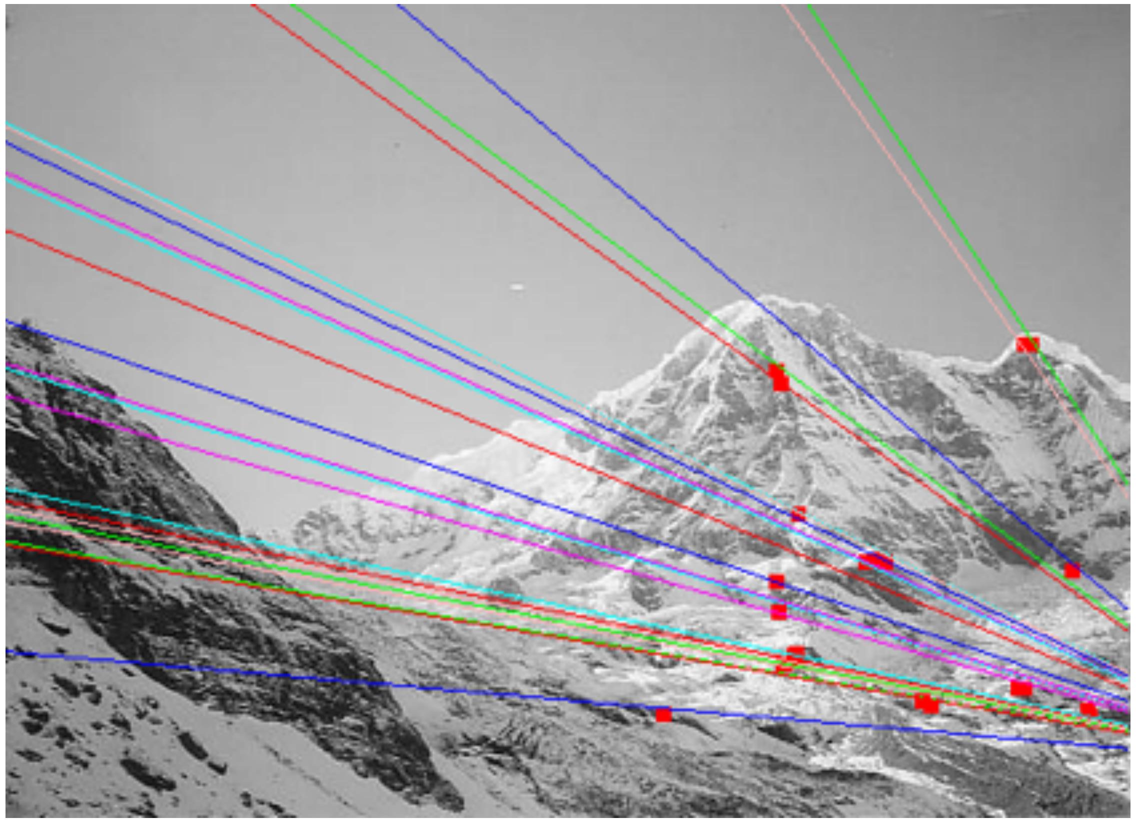
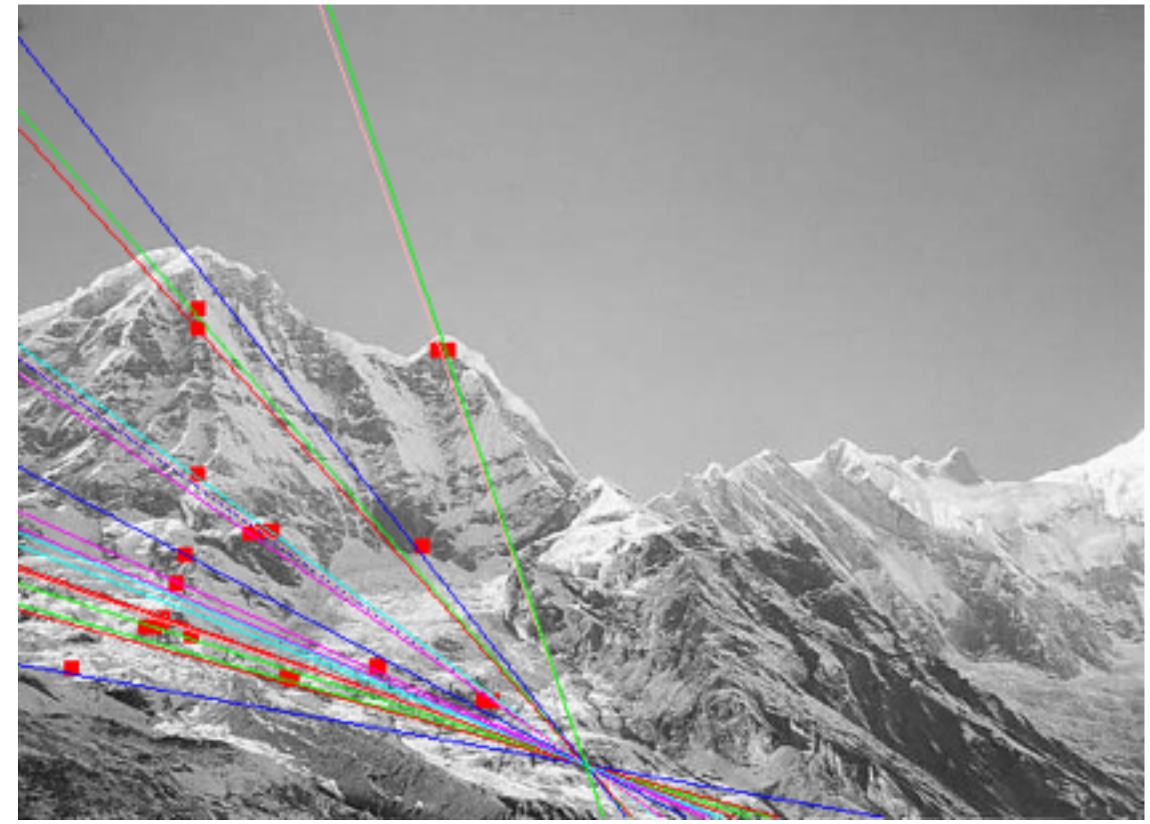
RANSAC w/ threshold=4, could be improved



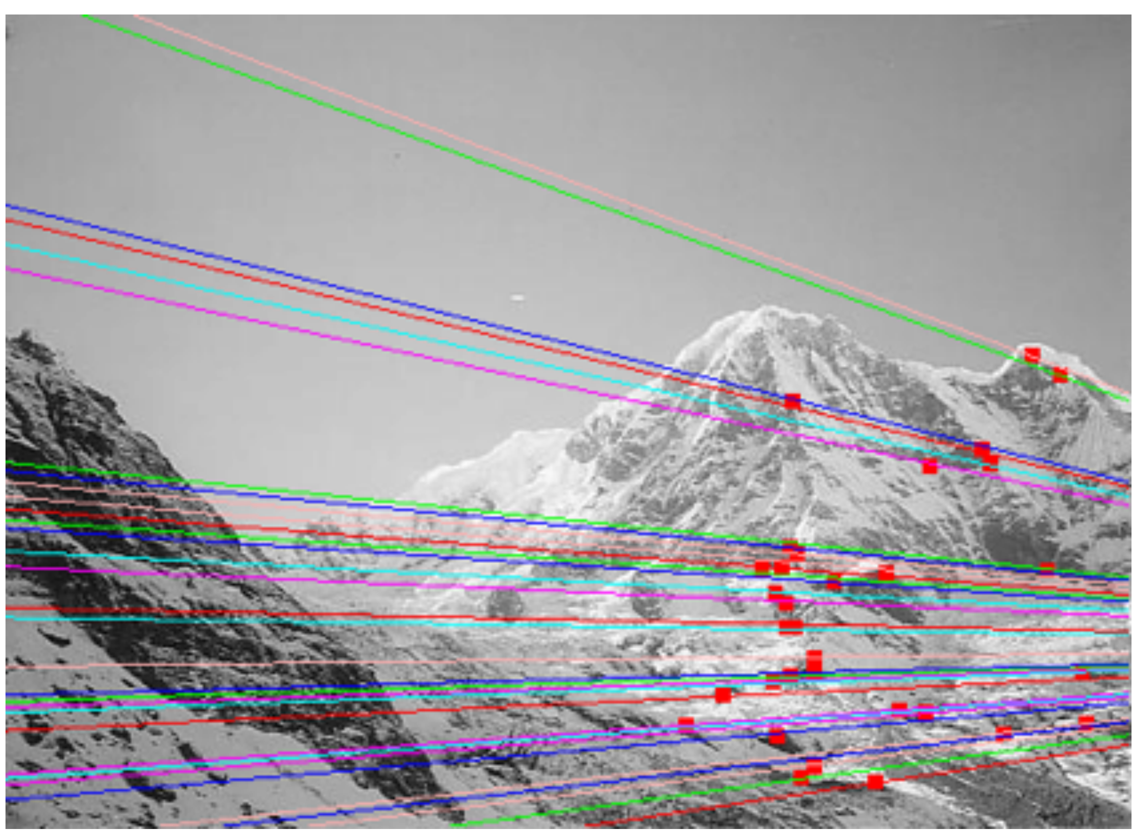
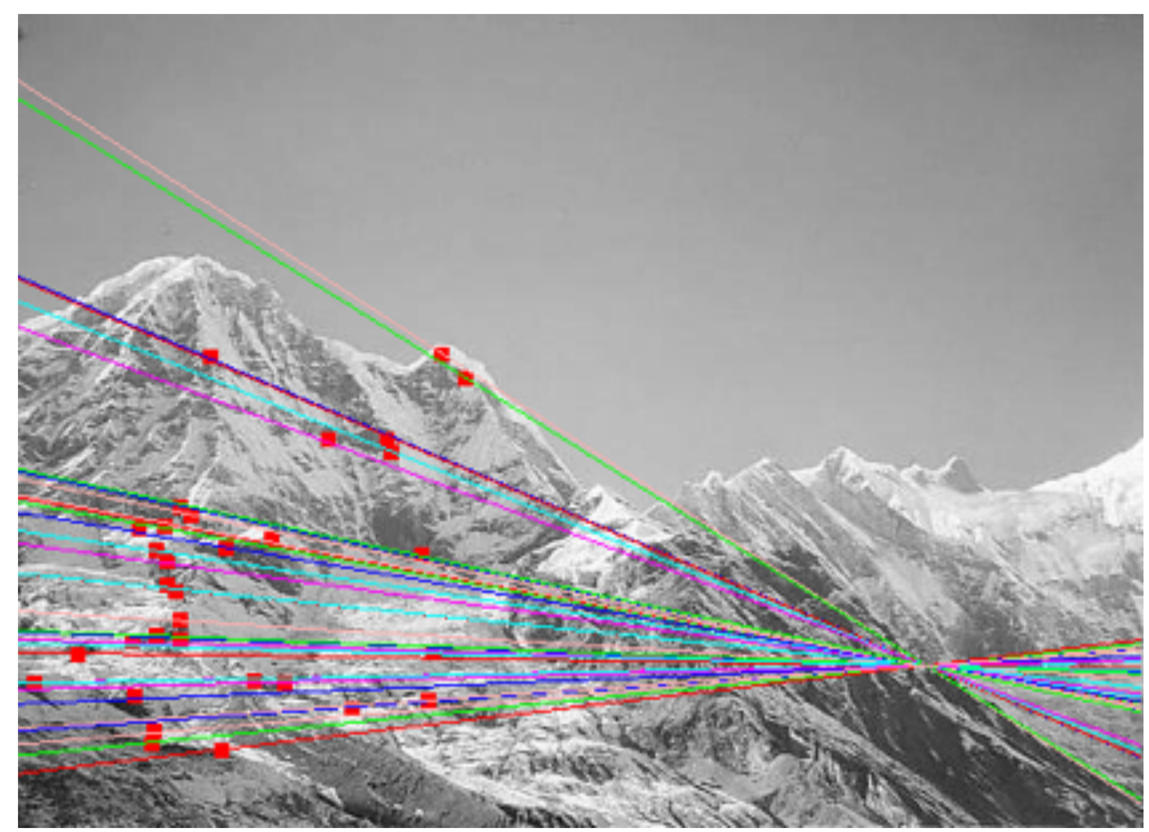
solution expected from manually matched:



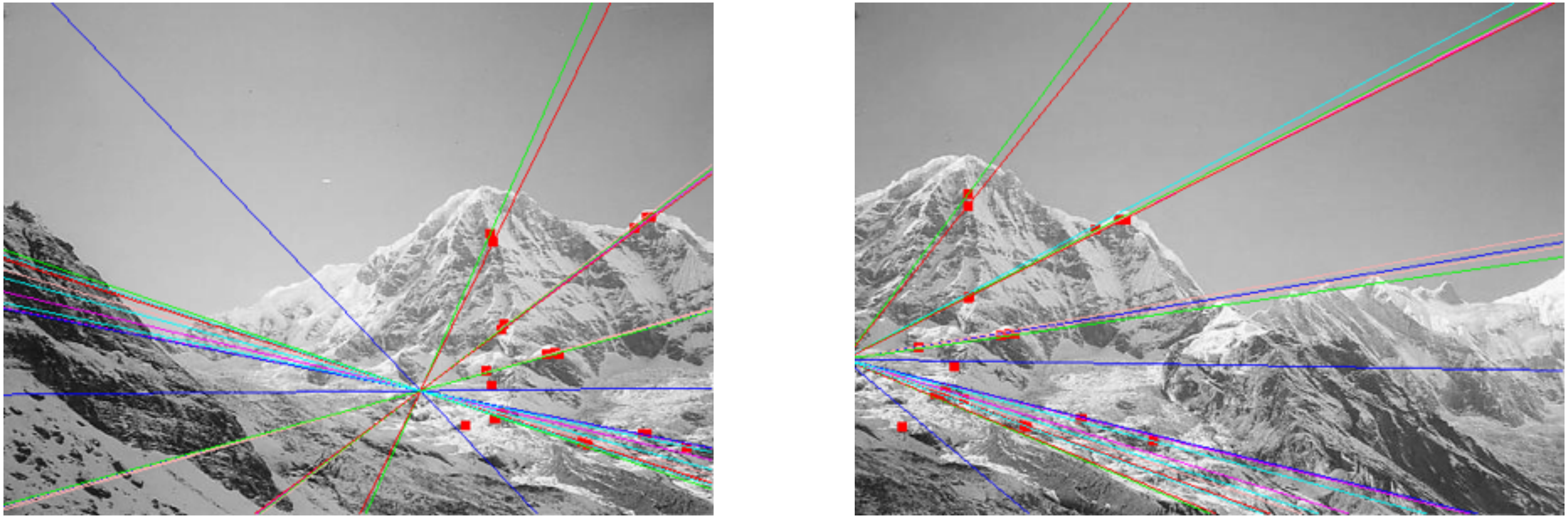
RANSAC w/ threshold=5, close (method is good. needs points at bottom.)

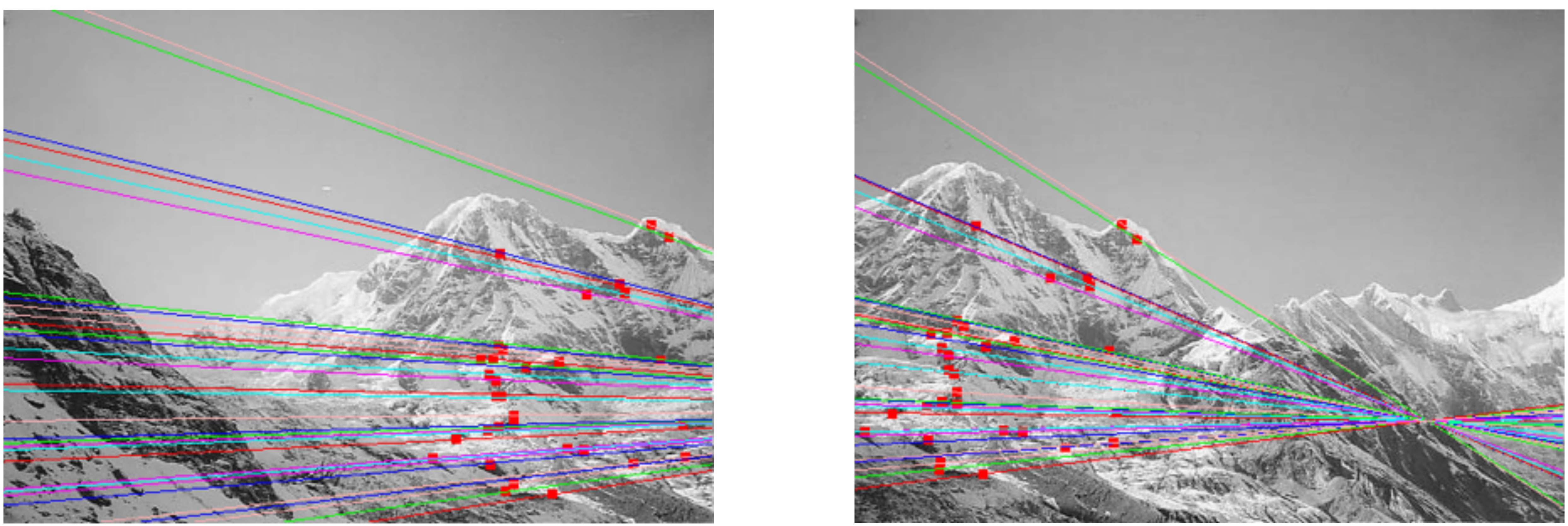
solution expected from manually matched:

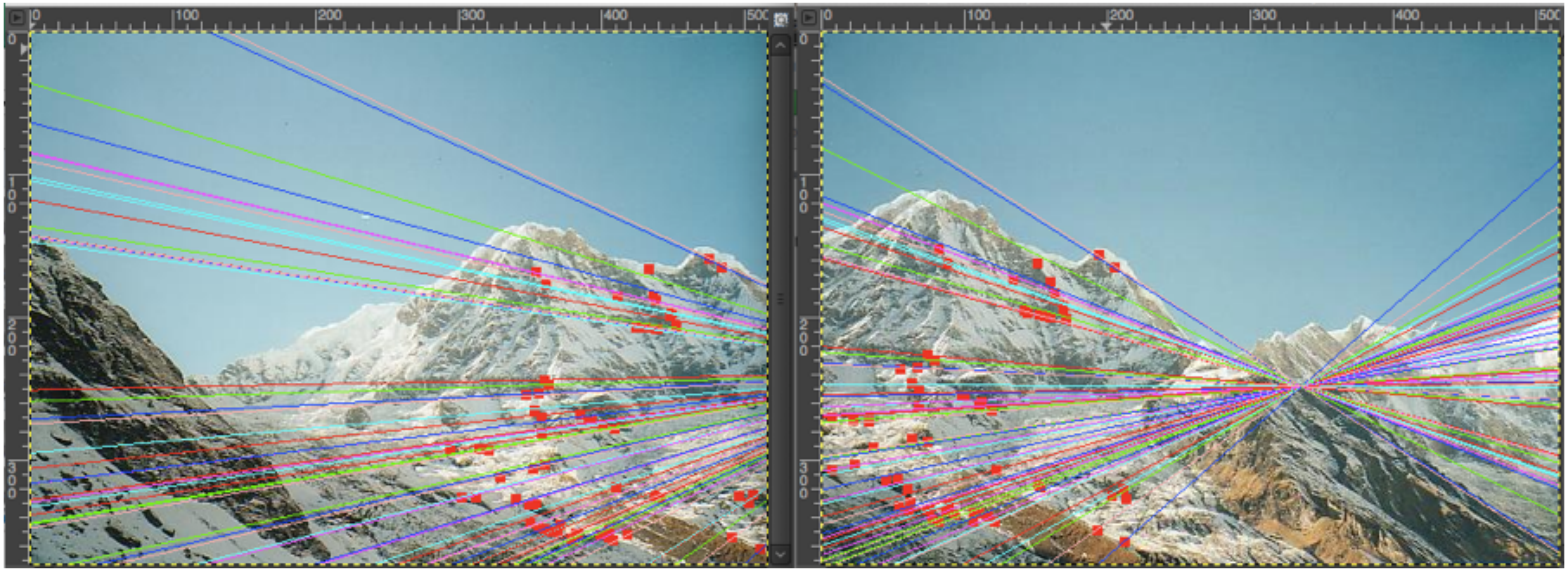
RANSAC w/ threshold=6, too large…5 was better

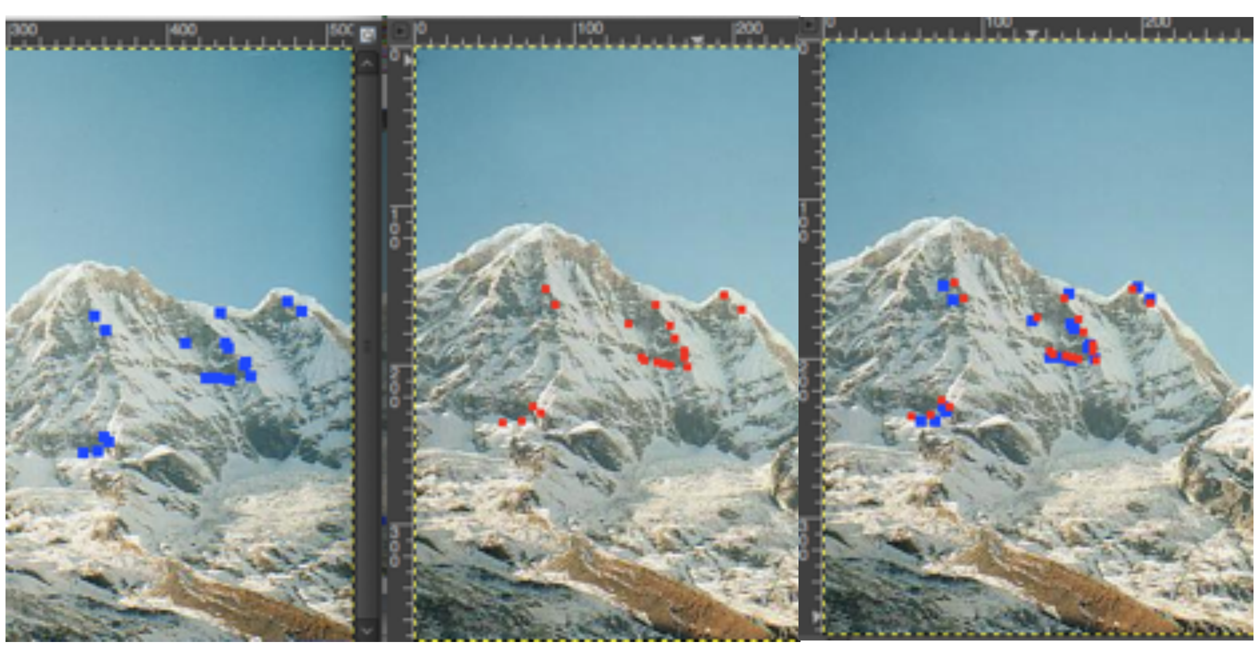


solution expected from manually matched:

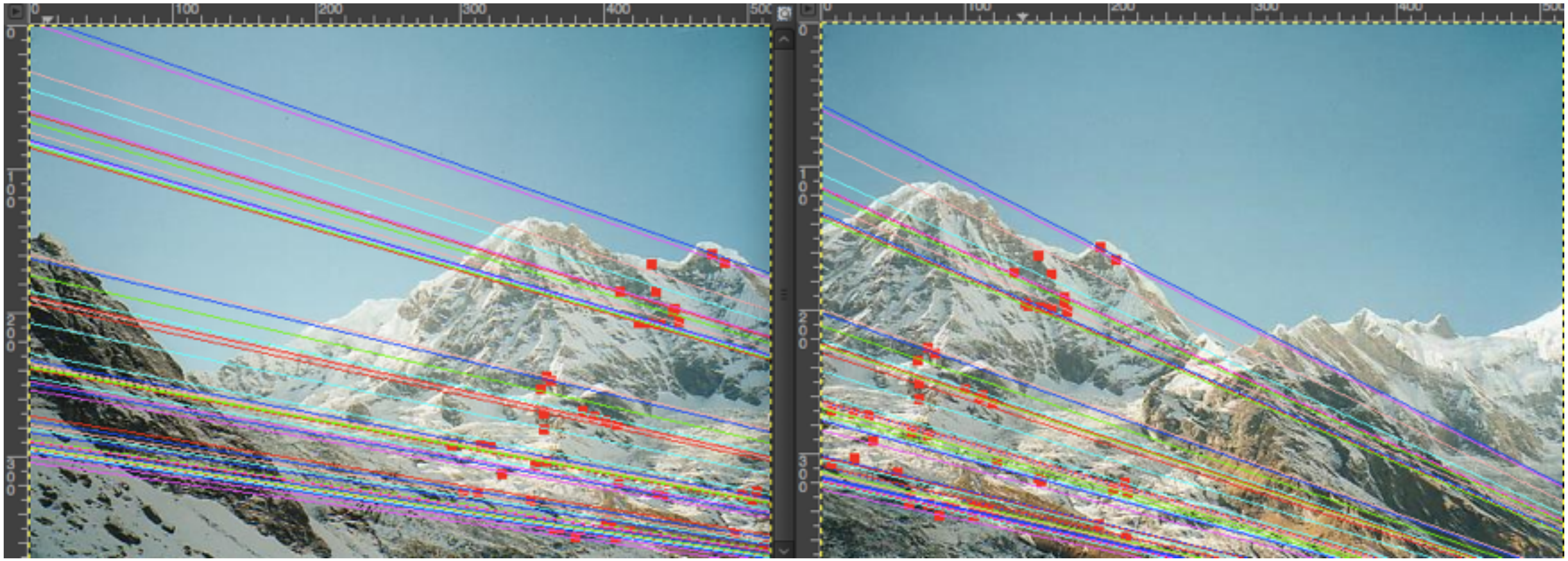


matching tol=7. final point matching is optimal for input into RANSAC. RANSAC tolerance = 5





matching tol=6. final point matching is optimal for input into RANSAC. RANSAC tolerance = 4



working on a 7-point method (update this)…

first, what the result should look like (using all points and the 8-point method) and then only 9: