```
points1 = moravec corner detection on image1 with threshold 100, maximum 40 points for 2% of image area, and discarding corners in patches with less than 6.0 shannon entropy

points2 = moravec corner detection on image2 with threshold 100, maximum 40 points for 2% of image area, and discarding corners in patches with less than 6.0 shannon entropy

image1 = image1 * 0.5

image2 = image2 * 0.5

binThreshold = 38

binNegationThreshold = 57

LOOP WHILE binThreshold < 56 AND (featureMatchCount < 5 OR featureMatchCloseness < 0.007)
    binMergeCount = 1

    binThreshold = binThreshold + 2

    binNegationThreshold = binNegationThreshold + 3

    reducedBinNegationThreshold = binNegationThreshold * 0.85

    LOOP featureMatchCount < 5 OR featureMatchCloseness < 0.007
        featureList1 = empty list

        LOOP points1
            concentricOvalList = empty list

            IF point x,y - 4 in points1 +/- radius 30 fits within image1 bounds
                outerHistogram = histogram of pixels within radius
                surroundedHistogram = histogram of pixels within radius / 3
                centralHistogram = histogram of pixels within radius / 7
                concentricOvals = [ outerHistogram, surroundedHistogram, centralHistogram ]
                concentricOvalList add concetricOvals

            IF point x - 4,y in points1 +/- radius 30 fits within image1 bounds
                outerHistogram = histogram of pixels within radius
                surroundedHistogram = histogram of pixels within radius / 3
                centralHistogram = histogram of pixels within radius / 7
                concentricOvals = [ outerHistogram, surroundedHistogram, centralHistogram ]
        concentricOvalList add concetricOvals

    IF point x + 4,y in points1 +/- radius 30 fits within image1 bounds
        outerHistogram = histogram of pixels within radius
        surroundedHistogram = histogram of pixels within radius / 3
        centralHistogram = histogram of pixels within radius / 7
        concentricOvals = [ outerHistogram, surroundedHistogram,
    centralHistogram ]
        concentricOvalList add concetricOvals

    IF point x,y + 4 in points1 +/- radius 30 fits within image1 bounds
        outerHistogram = histogram of pixels within radius
        surroundedHistogram = histogram of pixels within radius / 3
        centralHistogram = histogram of pixels within radius / 7
        concentricOvals = [ outerHistogram, surroundedHistogram,
    centralHistogram ]
        concentricOvalList add concetricOvals

    featureList1 add concentricOvalList

featureList2 = empty list

LOOP points2
    concentricOvalList = empty list

    IF point x,y - 4 in points1 +/- radius 30 fits within image2 bounds
        outerHistogram = histogram of pixels within radius
        surroundedHistogram = histogram of pixels within radius / 3
        centralHistogram = histogram of pixels within radius / 7
        concentricOvals = [ outerHistogram, surroundedHistogram,
    centralHistogram ]
        concentricOvalList add concetricOvals

    IF point x - 4,y in points1 +/- radius 30 fits within image2 bounds
        outerHistogram = histogram of pixels within radius
        surroundedHistogram = histogram of pixels within radius / 3
        centralHistogram = histogram of pixels within radius / 7
        concentricOvals = [ outerHistogram, surroundedHistogram,
    centralHistogram ]
        concentricOvalList add concetricOvals

    IF point x + 4,y in points1 +/- radius 30 fits within image2 bounds
        outerHistogram = histogram of pixels within radius
        surroundedHistogram = histogram of pixels within radius / 3
    centralHistogram = histogram of pixels within radius / 7
    concentricOvals = [ outerHistogram, surroundedHistogram, centralHistogram ]
    concentricOvalList add concetricOvals

IF point x,y + 4 in points1 +/- radius 30 fits within image2 bounds
    outerHistogram = histogram of pixels within radius
    surroundedHistogram = histogram of pixels within radius / 3
    centralHistogram = histogram of pixels within radius / 7
    concentricOvals = [ outerHistogram, surroundedHistogram, centralHistogram ]
    concentricOvalList add concetricOvals

featureList2 add concentricOvalList

LOOP featureList1
    LOOP concentricOvalList concentricOvals
        histogramLength = 256 / binMergeCount

        binIndex = 0

        angleIndex = 0

        sum1 = 0

        sum2 = 0

        distances1 = histogram of histogramLength

        LOOP histogramLength i
            sum1 += outerHistogram i

            sum2 += surroundedHistogram i

            binIndex = binIndex + 1

            IF binIndex = binMergeCount
                distances1 at angleIndex = sum1 - sum2

                angleIndex = angleIndex + 1

                binIndex = 0

        distances2 = histogram of histogramLength
```

```
sum1 = 0
sum2 = 0
binIndex = 0
angleIndex = 0
LOOP histogramLength i
    sum1 += surroundedHistogram i
    sum2 += centralHistogram i
    binIndex = binIndex + 1
    IF binIndex = binMergeCount
        distances2 at angleIndex = sum1 - sum2
        angleIndex = angleIndex + 1
        binIndex = 0
concentricOvals distances1 = distances1
concentricOvals distances2 = distances2
score = 0
LOOP outerHistogram i
    IF outerHistogram i < distinctiveness threshold 2
        score = score + 1
concentricOvals distinctiveness = 256 - score
concentricOvals max compare distinctiveness = concetricOvals distinctiveness + 10
concentricOvals min compare distinctiveness = concetricOvals distinctiveness - 10
longestSequence = 0
count = 0
```

```
        LOOP outerHistogram i
            IF outerHistogram i < 25
                count = count + 1

            ELSE
                IF longestSequence < count
                    longestSequence = count

                count = 0

        concentricOvals longestSequence = longestSequence

LOOP featureList2
    LOOP concentricOvalList concentricOvals
        histogramLength = 256 / binMergeCount

        binIndex = 0

        angleIndex = 0

        sum1 = 0

        sum2 = 0

        distances1 = histogram of histogramLength

        LOOP histogramLength i
            sum1 += outerHistogram i

            sum2 += surroundedHistogram i

            binIndex = binIndex + 1

            IF binIndex = binMergeCount
                distances1 at angleIndex = sum1 - sum2

                angleIndex = angleIndex + 1

                binIndex = 0

        distances2 = histogram of histogramLength

        sum1 = 0
```

```
sum2 = 0

binIndex = 0

angleIndex = 0

LOOP histogramLength i
    sum1 += surroundedHistogram i

    sum2 += centralHistogram i

    binIndex = binIndex + 1

    IF binIndex = binMergeCount
        distances2 at angleIndex = sum1 - sum2

        angleIndex = angleIndex + 1

        binIndex = 0

concentricOvals distances1 = distances1

concentricOvals distances2 = distances2

score = 0

LOOP outerHistogram i
    IF outerHistogram i < distinctiveness threshold 2
        score = score + 1

concentricOvals distinctiveness = 256 - score

concentricOvals max compare distinctiveness = concetricOvals distinctiveness + 10

concentricOvals min compare distinctiveness = concetricOvals distinctiveness - 10

longestSequence = 0

count = 0

LOOP outerHistogram i
```

```
            IF outerHistogram i < 25
                count = count + 1

            ELSE
                IF longestSequence < count
                    longestSequence = count

                count = 0

        concentricOvals longestSequence = longestSequence

distinctivenessModifier = 0.35

LOOP (featureList1 - count < 2500 AND featureList1 > 2500) OR first iteration
    sum = 0

    distinctivenessModifier = distinctivenessModifier - 0.05

    count = 0

    LOOP featureList1
        LOOP concentricOvalList concentricOvals
            sum = sum + concentricOvals distinctiveness

            count = count + 1

    sum = sum / count

    sumPiece = sum * distinctivenessModifier

    highSum = sum + sumPiece

    count = 0

    LOOP featureList1
        sum = 0

        LOOP concentricOvalList concentricOvals
            sum = sum + concentricOvals distinctiveness

        sum = sum / 4

        IF sum < highSum
```

```
            count = count + 1

LOOP featureList1
    sum = 0

    LOOP concentricOvalList concentricOvals
        sum = sum + concentricOvals distinctiveness

    sum = sum / 4

    IF sum < sumHigh
        featureList1 remove concentricOvalList

distinctivenessModifier = 0.35

LOOP (featureList2 - count < 2500 AND featureList2 > 2500) OR first iteration
    sum = 0

    distinctivenessModifier = distinctivenessModifier - 0.05

    count = 0

    LOOP featureList2
        LOOP concentricOvalList concentricOvals
            sum = sum + concentricOvals distinctiveness

            count = count + 1

    sum = sum / count

    sumPiece = sum * distinctivenessModifier

    highSum = sum + sumPiece

    count = 0

    LOOP featureList2
        sum = 0

        LOOP concentricOvalList concentricOvals
            sum = sum + concentricOvals distinctiveness

        sum = sum / 4
        IF sum < highSum
            count = count + 1

LOOP featureList2
    sum = 0

    LOOP concentricOvalList concentricOvals
        sum = sum + concentricOvals distinctiveness

    sum = sum / 4

    IF sum < sumHigh
        featureList2 remove concentricOvalList

LOOP featureList1 > 20000
    Remove concentricOvalList at random from featureList1

LOOP featureList2 > 20000
    Remove concentricOvalList at random from featureList2

LOOP featureList1
    LOOP concentricOvalList concentricOvals
        IF concentricOvals longestSequence > 70
            featureList1 remove concentricOvals

LOOP featureList2
    LOOP concentricOvalList concentricOvals
        IF concentricOvals longestSequence > 70
            featureList2 remove concentricOvals

featureMatchList = empty array

LOOP featureList1
    LOOP featureList2
        lowestDistance = 99999

        compareIndex = 0

        lowestRoughBinDistance = 99999

        LOOP [ [ 0, 1, 2, 3 ], [ 1, 2, 3, 0 ], [ 2, 3, 0, 1 ], [ 3, 0, 1, 2 ], ]
            compareIndex = compareIndex + 1
```

```
distanceFinal = 0

roughBinDistance = 0

IF feature1 distinctiveness < feature1 min distinctiveness OR feature1 distinctiveness > feature1 max distinctiveness
    distanceFinal = 99999

secondDistances1 = featureList2 concentricOvalList2 distances1

LOOP featureList1 concentricOvalList1 distances1 i
    val = distances1 i

    val2 = secondDistances1 i

    valLow = val * 0.98

    valThresholdCheck = | val - valLow |

    IF valThresholdCheck > 40
        valLow = val - 40

    valHigh = val * 1.02

    valThresholdCheckHigh = | val - valHigh |

    IF valThresholdCheckHigh > 40
        valHigh = val + 40

    IF val2 < valLow OR val2 > valHigh
        binDistance = binDistance + 1

        roughBinDistance = roughBinDistance + 1

        IF | val2 - val | < reducedBinNegationThreshold
            binDistance = binDistance - 1
        ELSE
            binDistance = binDistance + 1
            IF binDistance >= binThreshold
                BREAK LOOP

    secondDistances2 = featureList2 concentricOvalList2 distances2

    LOOP featureList2 concentricOvalList2 distances2 i
        val = distances2 i

        val2 = secondDistances2 i

        valLow = val * 0.98

        valThresholdCheck = | val - valLow |

        IF valThresholdCheck > 40
            valLow = val - 40

        valHigh = val * 1.02

        valThresholdCheckHigh = | val - valHigh |

        IF valThresholdCheckHigh > 40
            valHigh = val + 40

        IF val2 < valLow OR val2 > valHigh
            binDistance = binDistance + 1

            roughBinDistance = roughBinDistance + 1

            IF | val2 - val | < reducedBinNegationThreshold
                binDistance = binDistance - 1
            ELSE
                binDistance = binDistance + 1

            IF binDistance >= binThreshold
                BREAK LOOP

    distanceFinal = distanceFinal + binDistance

IF lowestDistance > distanceFinal
```

```
                        compareIndexMatch = compareIndex - 1

                        lowestDistance = distanceFinal

                        lowestRoughBinDistance = roughBinDistance

                distanceFinal = lowestDistance

                roughBinDistance = lowestRoughBinDistance

                IF distanceFinal < binThreshold
                    featureMatchList add feature match with roughBinDistance and compareIndexMatch

        featureMatchCount = featureMatchList size

        featureMatchCloseness = feature max x - min x * feature max y - min y / image1 width * height

        binMergeCount = binMegreCount + 1

    featureMatchCloseness = feature max x - min x * feature max y - min y / image1 width * height

index0Sum = 0

index1Sum = 0

index2Sum = 0

index3Sum = 0

LOOP featureMatchList match
    IF match compareIndexMatch = 0
        index0Sum = index0Sum + 1

    IF match compareIndexMatch = 1
        index1Sum = index1Sum + 1

    IF match compareIndexMatch = 2
        index2Sum = index2Sum + 1

    IF match compareIndexMatch = 3
        index3Sum = index3Sum + 1
```

```
maxIndexSum = MAX index0Sum index1Sum index2Sum index3Sum

maxIndex = index of maxIndexSum

LOOP featureMatchList match
    IF match compareIndexMatch IS NOT maxIndex
        featureMatchList remove match
```