The Cooling Rates of Two Dikes from Granite Falls. MN

Izaak Sunleaf, Matt Cotter, Marina Booth

Fall 2012

Introduction to Geology

Carleton College, Northfield, MN

Bereket Haileab

Introduction

This study will look at the crystallization patterns of two basaltic dikes from Granite Falls, MN (figures 1 and 2). Dikes are sheet-like, discordant igneous intrusions that cut vertically across country rock and are most commonly composed of diabase, basalt, or granitic rock. They range from a few centimeters in width to greater than thirty feet, and they are commonly of porphyritic texture (i.e., they contain larger crystals in a finer-grained matrix) therefore indicating more than one period of crystallization. As these magmatic intrusions rise from within the lower crust or upper mantle, they encounter progressively lower temperatures, causing partial crystallization.

Several past studies have concluded that the cooling rate of a dike affects both crystal size and mineral composition (e.g. Mollo et al. 2011; Webber et al. 1999). Our study is concerned with the former. In igneous rock formations, crystal size is positively correlated with cooling time. Therefore, the crystal size in a dike indicates the amount of time it took to cool depending on each crystal's distance from the contact of the dike and the country rock. Webber (1999) observed that variations in grain size in a dike are small and that grain sizes generally increase from dike margin to center.

The cooling rate of a dike is a valuable piece of information. According to Jaeger (1957), information about the temperatures under which a body is formed is needed in discussions of "metamorphism and rock magnetism, as well as of differentiation, grain size and the process of solidification of the igneous body."

The purpose of our project is to model the cooling rates of two different dikes, one dike 10-12mm wide and the other dike 106-108mm wide. We will model this based on the progression of their crystal sizes, and in so doing, verify that crystal size increases from the margin to the center of the dike. We will calculate the rate at which the grain size is changing, and subsequently, the cooling rates of the two dikes.

Methods

Figure 2: the 10-12mm dike and its corresponding thin section slides. In both figure1 and 2, the lighter country rock bounds the darker, basaltic dike.

We started by acquiring thin sections of our two different dikes from Professor Bereket Haileab. We continued by examining them underneath a petrographic microscope. We next took photos of the slides at different distances from the dike-country rock contact (see figures 3-20). We measured the increments of distances on the slide between each image. With the images and various computer applications (including Microsoft Excel, Adobe Photoshop, and our own program), we found the area of the feldspar and pyroxene crystals within each slide. To isolate the crystals within the image, we used Adobe Photoshop. Taking advantage of the luster of the feldspars and pyroxenes, we were able to use Adobe Photoshop to recognize the crystals. Using the Photoshop color range select function, we selected the highlighted or brighter pixels, thus identifying the crystals in the image. We then copied only the highlighted pixels and pasted them into a new image with a black background to more clearly see the crystals.

After doing this, we created an algorithm using Processing (a library built on Oracle's Java) to detect crystals from the highlighted pixels and then count and analyze them (see figure 25). The image was converted to black and white for the purpose of distinguishing crystals from the image's background. Each individual crystal was detected by grouping contiguous pixels. After creating a collection of adjacent pixels, the program recorded the distance from the center of each crystal to the left-side edge of the image and the size of each crystal (both measured in pixels). All of this data was imported into Microsoft Excel. We converted the units of size and distance from pixels to microns using the scale provided on each image by the petrographic microscope. Next we used Microsoft Excel to plot all of the data gathered from each slide. We then used an exponential regression analysis to determine a function modeling crystal size verses distance from the edge of the dike (see figures 21-24). Because crystal size directly correlates with the cooling time, we then used our functions to determine a formula for the cooling rate of each dike.

Results

Upon our initial observation, we noticed that the crystal size increases with increasing distance from the country rock at a non-linear rate. This is supported by readings that state that the magma closer to the country rock cools faster than magma closer to the center of the dike due to the proximity to the much colder country rock (e.g., Mollo et al. 2011; Jaeger 1957). Also, we noticed that the crystals aligned parallel to the contact of the country rock near the edge of the dike. The largest crystals that formed in the larger dike were significantly greater in size than the largest crystals that formed in the smaller dike by approximately 400%. This implies that the dike's width is also a factor in the cooling rate of the dike.

Our numerical data also show some promise. The graphs support our hypothesis (see figures 21-24). Looking at the graphs, the crystal sizes clearly get larger at farther distances from the country rock, thus supporting that the crystals closer to the middle of the dike cool at a slower rate than the crystals closer to the edge of the dike. The graphs also demonstrate a clear shape that looks like a polynomial graph, possibly to the second power. However, for a number of reasons the actual calculated equation for the line did not turn out well. Some of the graphs don't have any large crystals represented in some of the images. There are so many smaller crystals that the calculated trend line is almost nearly a line with no slope; however, this can also be attributed to the thickness of the dikes that we were analyzing (the wider dike is 106-108mm and the thinner is 10-12mm). We could attain a larger data set with a larger dike and more time, which could then demonstrate a wider variety in crystal size especially in larger crystals. The Photoshop highlighting works very well, but it still can confuse and break up crystals which gives a larger number of smaller crystals, bringing down the trend line to further miscalculate the cooling rate, furthermore, this can potentially select very small pieces of the original image that aren't actual crystals. Overall, the data gave us promising results that could be explored further.

Discussion

The relationship between the cooling rate of the feldspar and pyroxene crystals and the distance from the dike-country rock contact is clear but not exact. The relationship is demonstrated through shape of the graph and the size of the crystals in the photographs. The number of small crystals and the imprecise nature of Photoshop and our methods made it difficult to determine the exact cooling rate with the tools at hand, but it did give us an estimation. With more materials, larger dikes and a refined methodology, we can replicate a more precise experiment with better data that could lead us to an actual equation. As noted earlier, one of the dikes we examined had a maximum width scarcely larger than 10mm. A wider dike would give us a wider variety of crystal sizes because the crystals farther from the dike-country rock edge would have even more time to develop – the study done by Jaeger (1957) suggests that the dike cools at a rate proportional to the square of its width. To address the numerous smaller crystals, our current minimum threshold for the area of a crystal is 5 pixels. Increasing this to a much larger number (one that could be found empirically through selective measurements in existing images) might result in an equation for the crystal size more representative of the cooling rate. While this would require more effort, an actual equation for the cooling rate would be very useful because it would give geologists the ability to estimate crystal size and ultimately prove (by this methodology) that the dikes cool more slowly in the middle than near the edge.

There are a few other aspects of our research that would be interesting to address further. The thin sections of the slides only give us a cross-section area of the crystals rather than a volume. We could make thin sections of the dike at different angles and develop a way to calculate the volume to make our cooling rate calculations more accurate. Another interesting thing we noticed in our images is that the feldspar crystals closer to the margin of the dike are parallel to the contact between the dike and country rock. This pattern leads us to believe that the crystals developed further down in the dike and the magma flow carried them up. This creates the questions: where did the crystals form, and did all of the crystals within a given cross-section form at the same time and in their current orientation? This presents an additional layer of complication to determining an equation for the cooling rate of the dikes. These new issues add another dimension to our research that would be very interesting to investigate in further studies.

Conclusion

Both dikes that we examined cooled at a faster rate at the dike-country rock contact than in the center of the dike. Due to the quantity of small crystals in our data, the exact cooling rate of the dikes was difficult to find, however there is some evidence to support this hypothesis even in the smaller dike and substantial quantitative support in the larger dike. Another trend we observed in this study was that the number of plagioclase feldspar crystals increased significantly with distance from the dike's contact with the country rock.

References

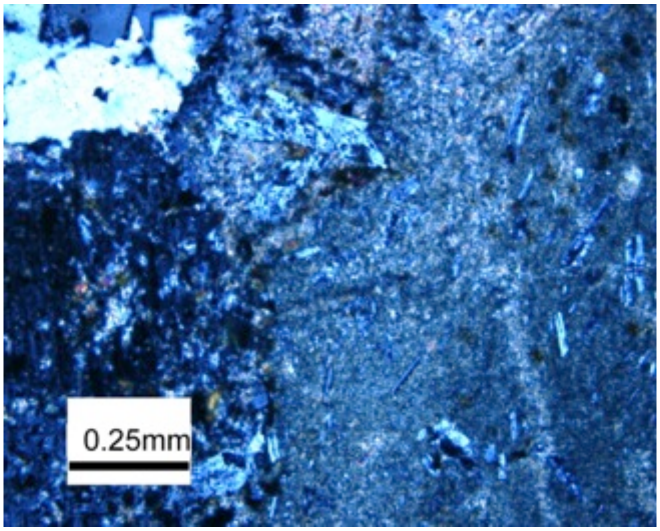
Jaeger, J.C., 1957, The temperature in the neighborhood of a cooling intrusive sheet: American Journal of Science, v. 255, p. 306–318.

Karson, J.A., 2002, Geologic Structure of the Uppermost Oceanic Crust Created at Fast- to Intermediate-Rate Spreading Centers: Earth Planet Science, v. 30, p. 347-384.

Mollo, S., Lanzafame, G., Masotta, M., Iezzi, G., Ferlito, C., Scarlato, P., 2011, Cooling history of a dike as revealed by mineral chemistry: A case study from Mt. Etna volcano: Chemical Geology, v. 288, p. 39-52.

Snelling, A.A., 1991, The formation and cooling of dikes: Creation Ex Nihilo Technical Journal, v. 5, 81-90.

Webber, K.L., Simmons, W.B., Falster, A.U., Foord, E.E., 1999, Cooling rates and crystallization dynamics of shallow level pegmatite-aplite dikes, San Diego county, California: American Mineralogist, v. 84, p. 708-717.



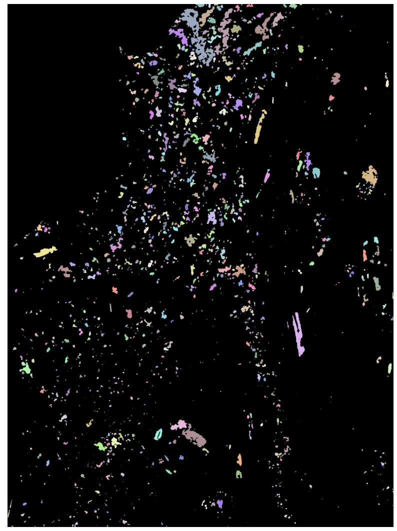
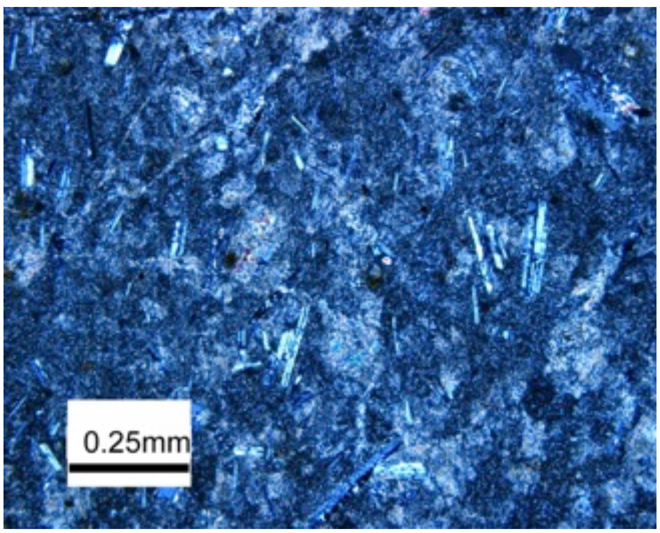


Figure 3 (left) and 4 (right): slide 3DM-26_57mm



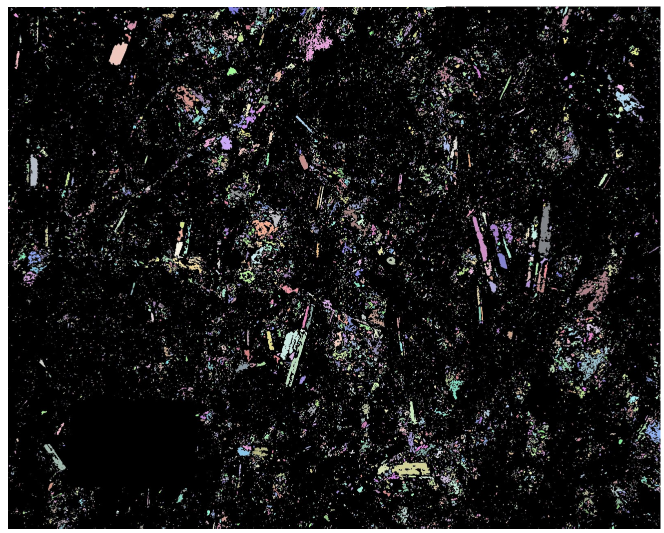
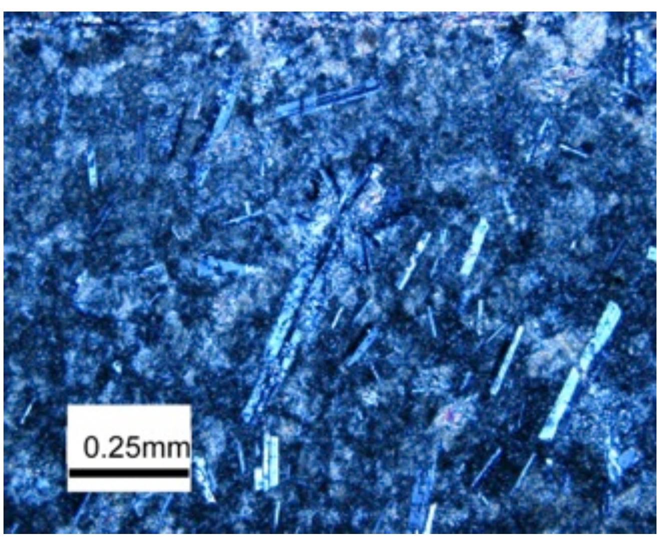


Figure 5 (left) and 6 (right): slide 3DM-26_59mm



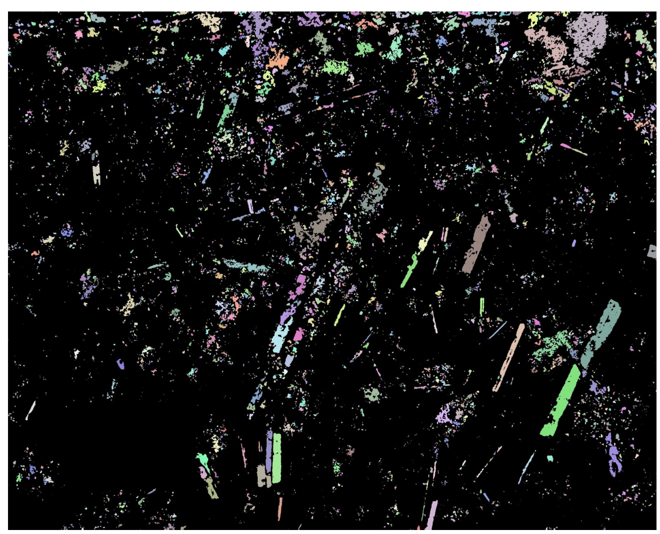
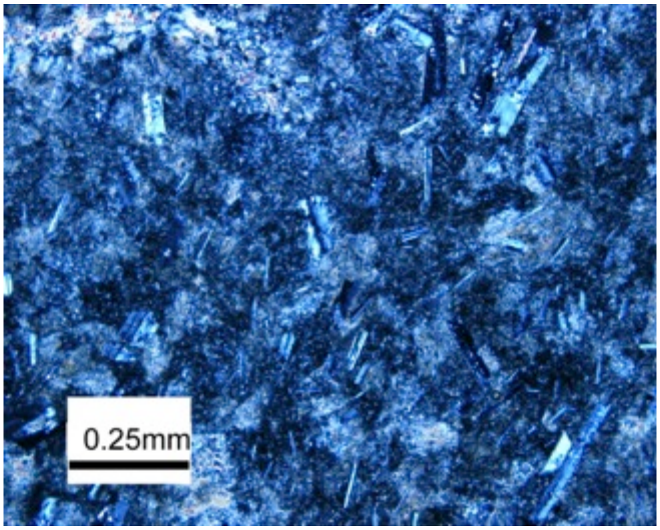


Figure 7 (left) and 8 (right): slide 3DM-26_61mm



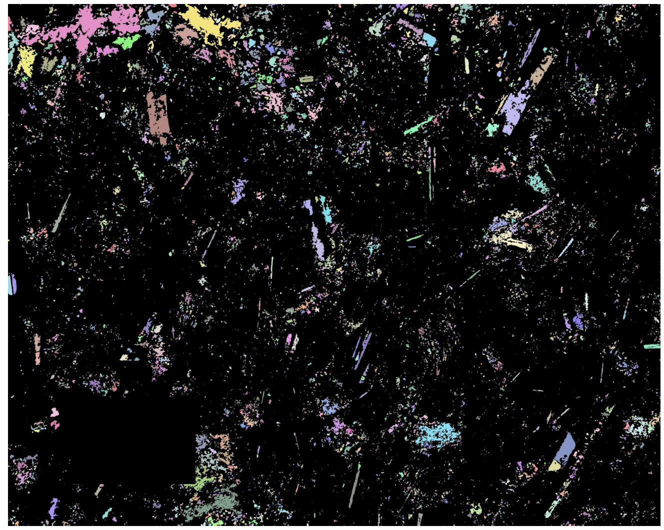
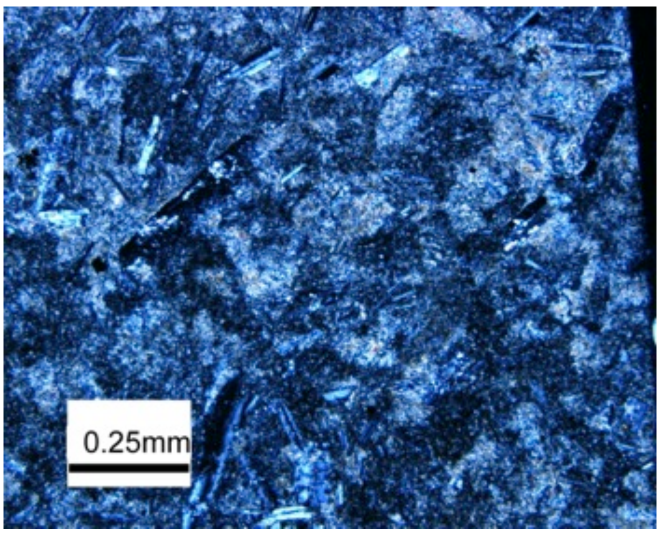


Figure 9 (left) and 10 (right): slide 3DM-32_10mm



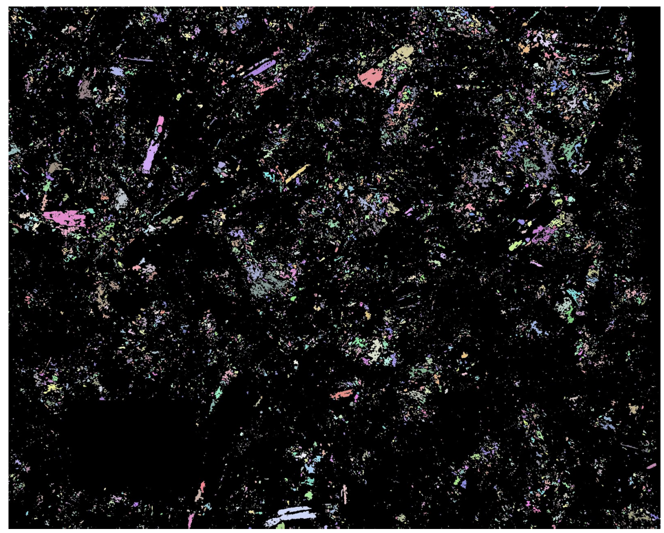
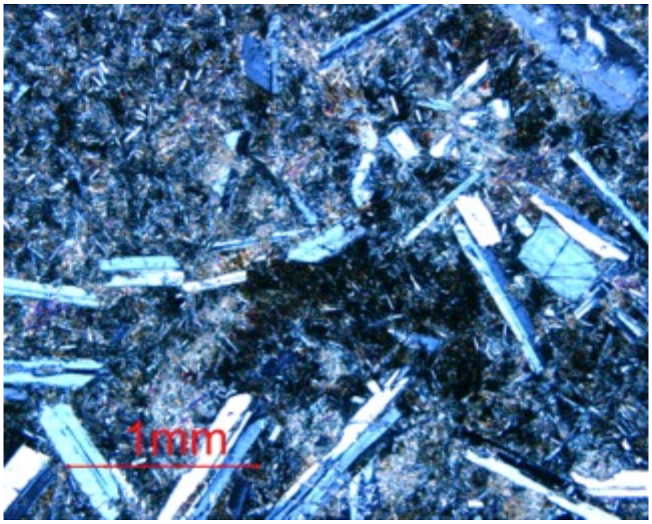


Figure 11 (left) and 12 (right): slide 3DM-26_65mm



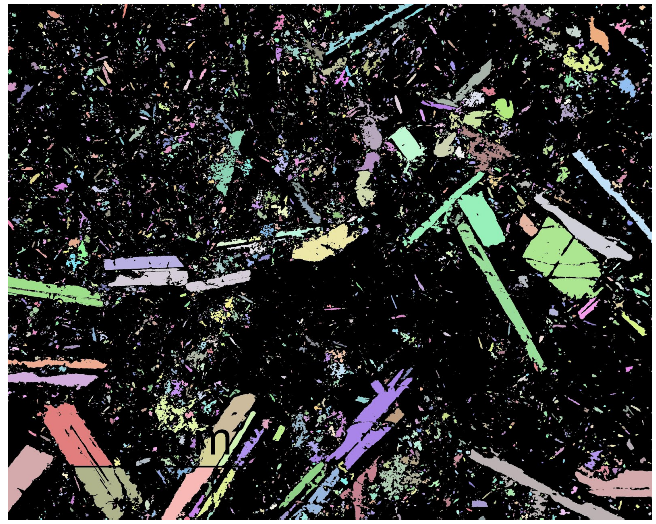
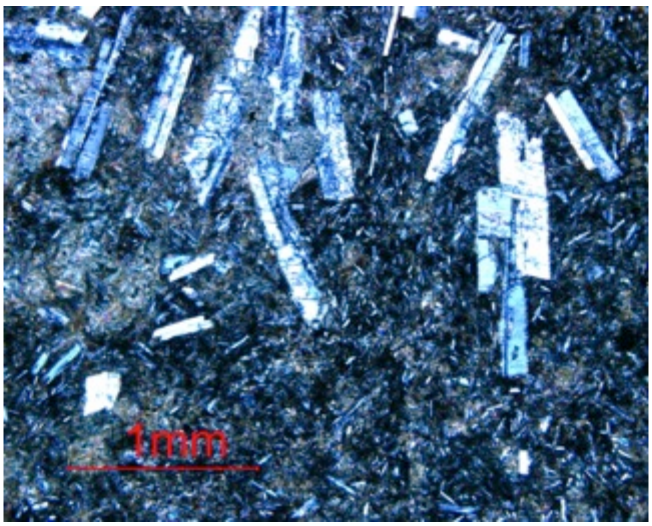


Figure 13 (left) and 14 (right): slide 3DM-32_10mm



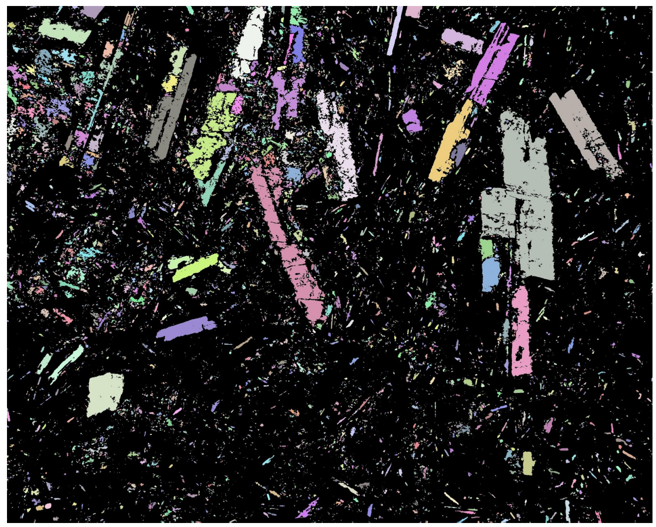
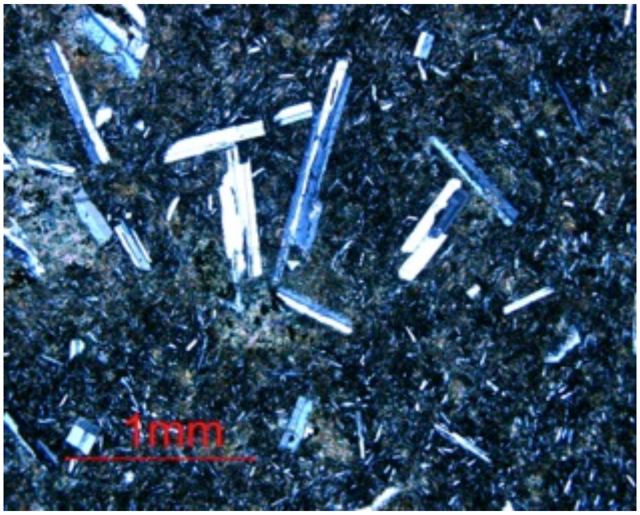


Figure 15 (left) and 16 (right): slide 3DM-32_23mm



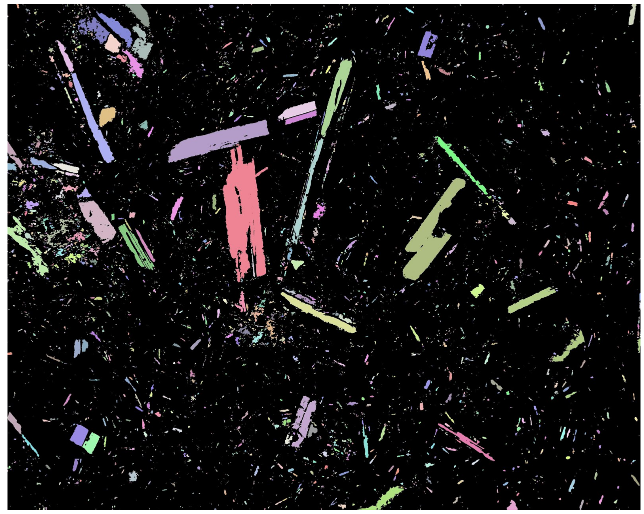
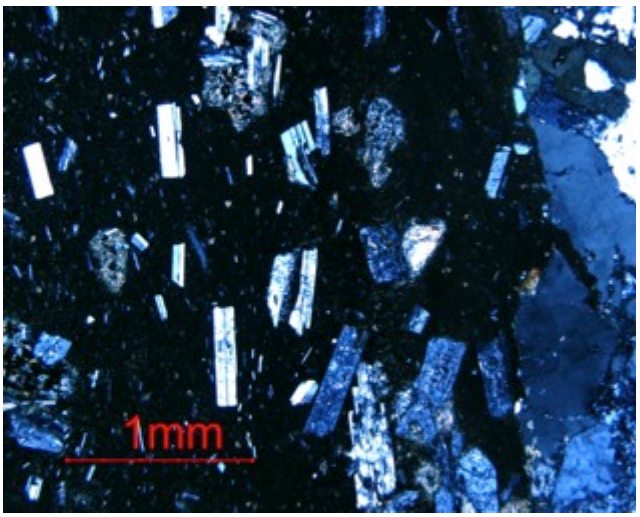


Figure 17 (left) and 18 (right): slide 3DM-32_41mm



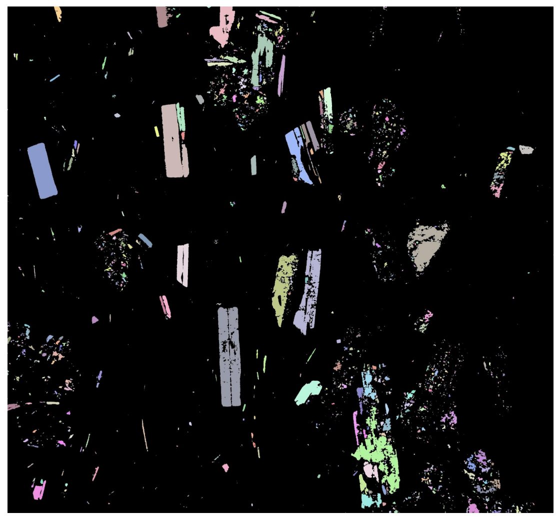
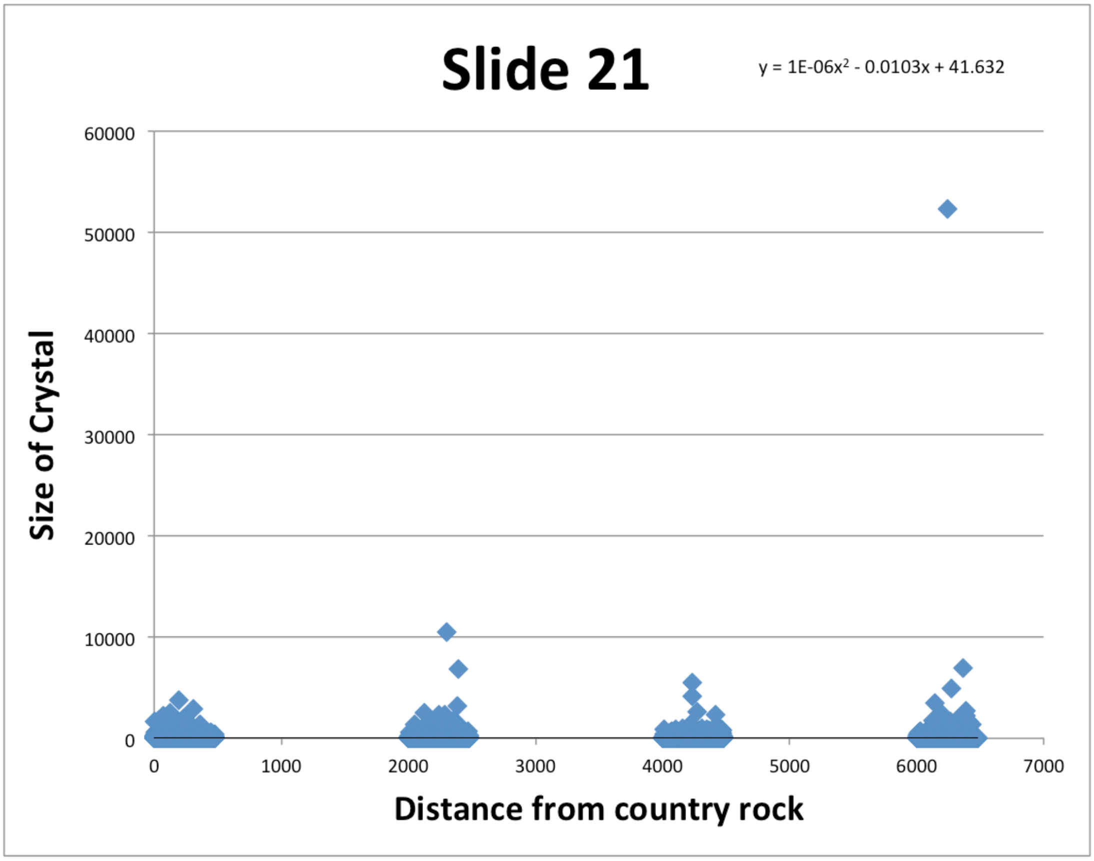


Figure 19 (left) and 20 (right): slide 3DM-32_59mm



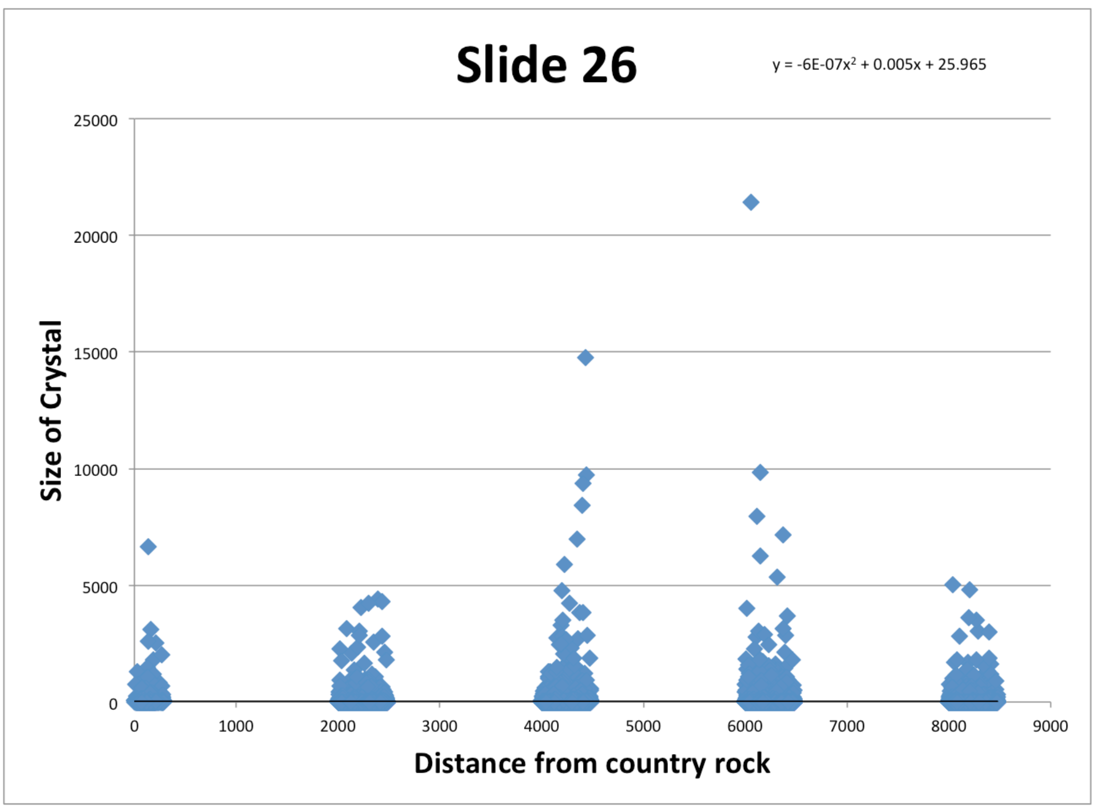
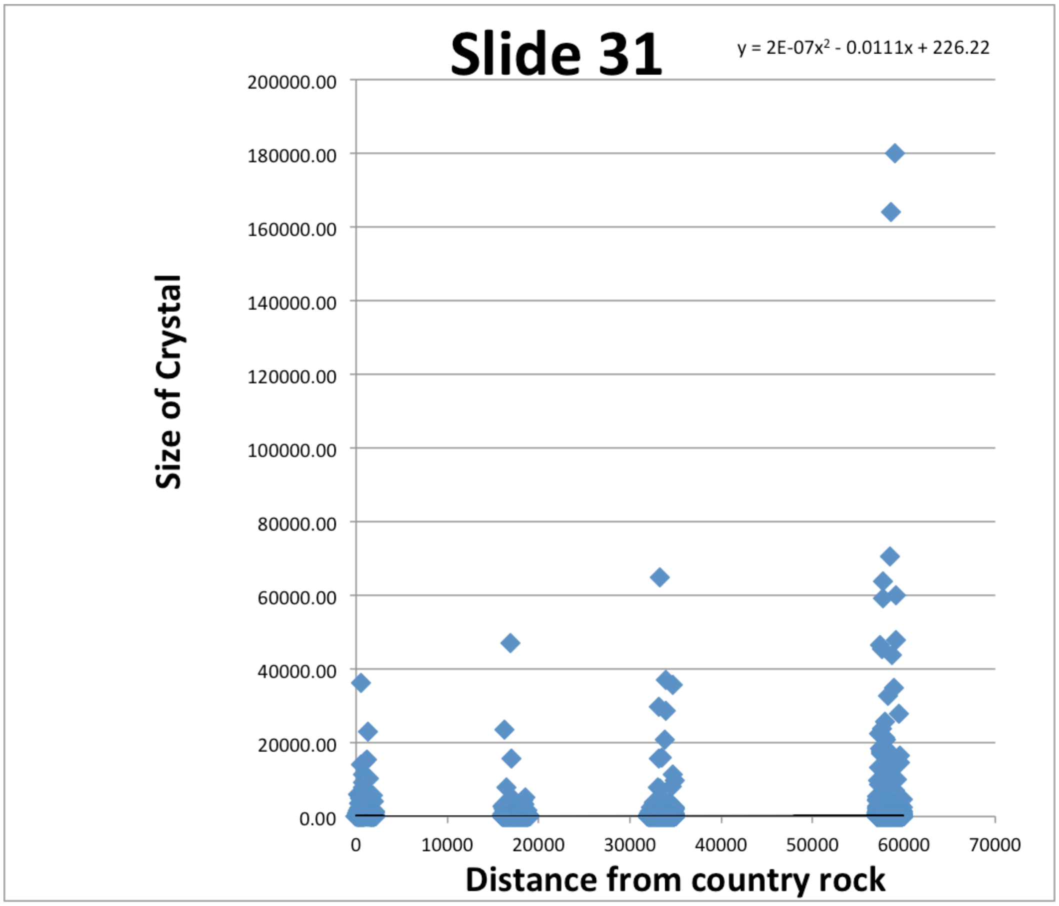


Figure 21 (top) and 22 (bottom): results given by program for Dike 2.



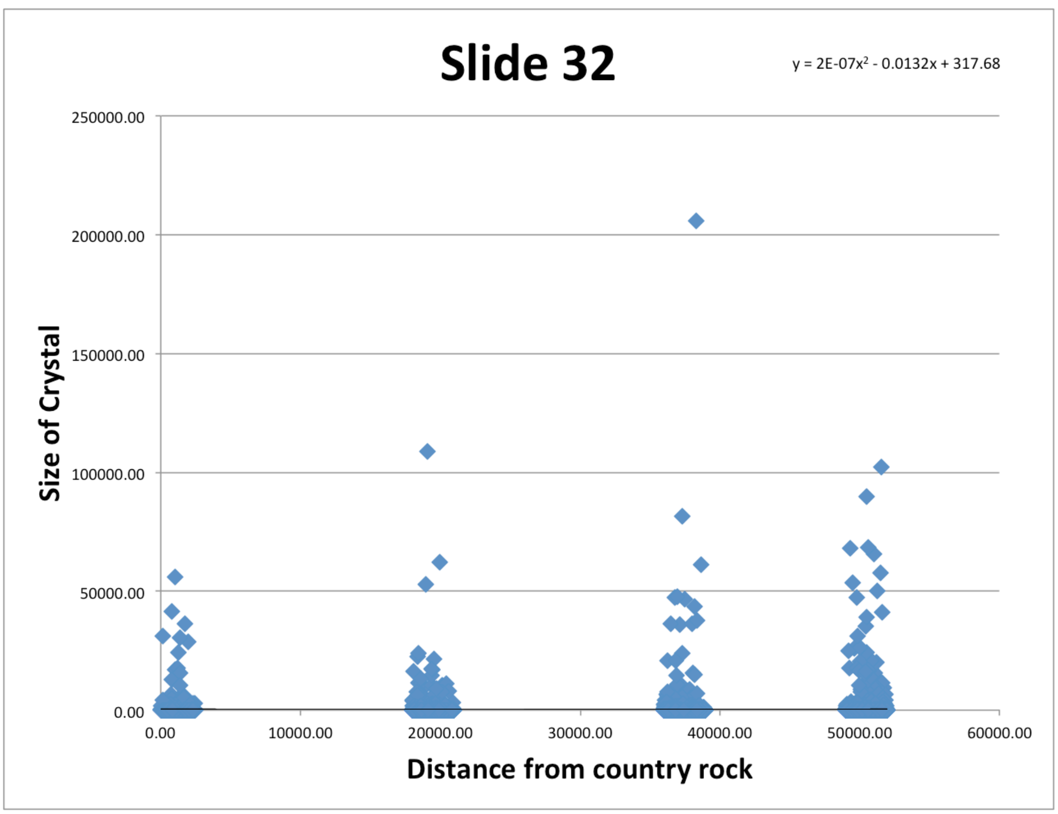


Figure 23 (top) and 24 (bottom): results given by program for Dike 1.

Figure 25: program code

```
int minCrystalSize = 5;
int greyVal = 65;
color grey = color(greyVal, greyVal, greyVal);
color black = color(0, 0, 0);
color white = color(255, 255, 255);

//Convert image to Black and White
PImage toBW(PImage img, String fname) {
  PImage imgCopy = img.get();
  imgCopy.loadPixels();
  int[] pix = imgCopy.pixels;
  for (int i=0; i<pix.length; i++) {
    if (pix[i]>grey) {
      pix[i] = white;
    }
    else {
      pix[i] = black;
    }
  }
  imgCopy.updatePixels();
  imgCopy.save(fname);
  PrintWriter output = createWriter(fname.replace("bw.jpg","pixCount.txt"));
  output.println("Crystal pixels: "+countWhite(imgCopy));
  output.flush();
  output.close();
  return imgCopy;
}

//Counts how many pixels of the given
//    color occur in the given image
int countCol(PImage img, color c) {
  int[] pix = img.pixels;
  int count = 0;

  for (int i=0; i<pix.length; i++) {
    if (pix[i]==c) {
      count++;
    }
  }
  return count;
}


//Counts how many white pixels occur in the given image
int countWhite(PImage img) {
  return (countCol(img,white));
}

color randCol(){
  return color(int(random(125)+125),int(random(125)+125),int(random(125)+125));
}
```

```
PImage anylizeCrystals(PImage Img,String outFName){
  PImage img = Img.get();
  img.loadPixels();
  int[] pix = img.pixels;
  PrintWriter output = createWriter(outFName);
  output.println("Crystal Size (in pixels),Distance from edge of picture to center of crystal");
  for (int i=0; i<pix.length; i++) {
    if (pix[i]==white) {
      P startingP = new P(i % img.width,i / img.width);
      CrystalBuilder cb = new CrystalBuilder();
      color col = randCol();

      //recursive function initial call
      crystalFind(pix, cb, startingP, col, img.width);

      CrystalData cd = cb.build();
      if(cd.pixCount>=minCrystalSize){
        output.println(cd.pixCount+","+cd.center);
      }
    }
  }
  output.flush();
  output.close();
  img.updatePixels();
  return img;
}

int getPixIndex(P p, int imgWidth){
  return (p.x + p.y*imgWidth);
}


void crystalFind(int[] pix, CrystalBuilder cb, P p, color col, int imgWidth){
  int pIndex = getPixIndex(p,imgWidth);

  try{
    //return if this is not a white pixel
    if (pix[pIndex] != white){ return;}
    //or return if we are outside the array bounds.
  }catch(Exception E){return;}

  pix[pIndex]=col;
  cb.addPoint(p);
  boolean looper = true;
  int radius = 1;
  while(looper){
    looper=false;
    for(int i=(-1*radius); i<=radius; i++){
      for(int j=(-1*radius); j<=radius; j++){
        int currCord = pIndex+i+(imgWidth*j);
        if((currCord<0)||(currCord>=pix.length)||(pix[currCord]!=white)){continue;}
        if(isAdjacentToo(pix,imgWidth,currCord,col)){
          looper=true;
          pix[currCord]=col;
          cb.addPoint(currCord % imgWidth, currCord / imgWidth);
        }
```

```
      }
    }
    radius++;
  }
}

boolean isAdjacentToo(int[] pix, int imgWidth, int index, color col){
  for(int i=-1; i<2; i++){
    for(int j=-1; j<2; j++){
      int currCord = index+i+(imgWidth*j);
      if((i==0 && j==0)||(currCord<0)||(currCord>=pix.length)){continue;}
      if(pix[currCord]==col){return true;}
    }
  }
  return false;
}

void setup() {
  String[] fileList = {"Slide3DM-21(1)_60mm","Slide3DM-21(1)_62mm","Slide3DM-
21(1)_64mm","Slide3DM-21(1)_66mm",
      "Slide3DM-26(20)_57mm","Slide3DM-26(20)_59mm","Slide3DM-26(20)_61mm","Slide3DM-
26(20)_63mm","Slide3DM-26(20)_65mm",
      "Slide3DM-31_22mm","Slide3DM-31_38mm","Slide3DM-31_54mm","Slide3DM-31_79mm",
      "Slide3DM-32_10mm","Slide3DM-32_23mm","Slide3DM-32_41mm","Slide3DM-32_59mm"};
  for(String rockName:fileList){
    println("\nAnalyzing "+rockName);

    println("Loading image...");
    PImage rockTest = loadImage(rockName+".jpg");
    println("Image loaded.\n");

    println("Converting to black and white...");
    PImage bw = toBW(rockTest, "results/"+rockName+"/bw.jpg");
    println("Image converted.\n");

    println("Analyzing crystals...");
    PImage analyzed = anylizeCrystals(bw,"results/"+rockName+"/data.csv");
    analyzed.save("results/"+rockName+"/highlighted.jpg");
    println("Crystals analyzing./n");
  }
  println("\nComplete!");
  exit();
}
```

```
public class P{
  public int x,y;
  public P(int x, int y){
    this.x=x;
    this.y=y;
  }
  public P(P p){
    this.x=p.x;
    this.y=p.y;
  }
}
```

```java
public class CrystalData{
  public int center,pixCount;
  public CrystalData(int c, int p){
    this.center = c;
    this.pixCount = p;
  }
}
```

```java
public class CrystalBuilder{
  ArrayList<P> crystalPoints;

  public CrystalBuilder(){
    crystalPoints = new ArrayList<P>();
  }

  public void addPoint(int x, int y){
    crystalPoints.add(new P(x,y));
  }

  public void addPoint(P p){
    crystalPoints.add(new P(p));
  }

  public CrystalData build(){
    int[] across = new int[2];
    across[0]=Integer.MAX_VALUE;
    across[1]=Integer.MIN_VALUE;
    for(P p:crystalPoints){
      if(p.x<across[0]){
        across[0]=p.x;
      }else if(p.x>across[1]){
        across[1]=p.x;
      }
    }
    return new CrystalData((across[0]+across[1])/2,crystalPoints.size());
  }

}
```