Vector Mapping Stitching

Roll no.:BS2006

August 23, 2023

(clickable content)

Contents

1 Objective

We aim to create a map of an area with some satellite images. We have with us some satellite images which cover a large area together. All these satellite images overlap with at least one other image. We mark points of interest in all the images, some of which are common in multiple images. From these common points, we estimate the relative positions of all points of interest with respect to one reference point which is fixed earlier. Finally, we join the points by roads to get a map of the area.

2 Data Collection:

My hometown is Tufanganj, in West Bengal, India. I took 6 satellite images fixing the resolution from Google Maps of the North-West part of the town. I loaded the screenshots one by one in a unit length square plot window whose bottom-left corner is at (1,1) and top-right corner at (2,2) in R software using my R-script. Then it (the script) took user-input of what is screenshots no. and then it asks the name of the point when a point is located using the locator (which collects the x-coordinate and y-coordinate) and saves all this information. I took the screenshots such that they share some common parts which contain a certain no. of points because this points are going to determine the underlying relation between the screenshots. So, I got repeated data from different screenshots for several points. No. of unique places of which data is collected is 21.

3 Model Description

Next, we need to fit a linear model to this to estimate relative positions. We can consider x and y coordinates independently since they don't have any effect on each other. So, we can estimate them separately and combine them at the end to get the final positions. Say X_{ij} is the x-coordinate of the ith location in the jth picture. Say we have a fixed reference point and scale. X_ij will be dependent on the true position of the location (noted by its name) and the picture from where it is taken. We can also have some random errors while clicking the points and the satellite images might also have some small random errors. I considered Name and Pic as factor variables here. No interaction term is expected since they're independent. So, I considered the model as:

$$x_ij = Name_i + Pic_j + \epsilon$$

y_ij can be modelled in a similar way. The design matrix of this model will have rank:(Names+Pics-1) since each name and pic is a factor and the '-1' because Pic1 is not estimable, it is taken

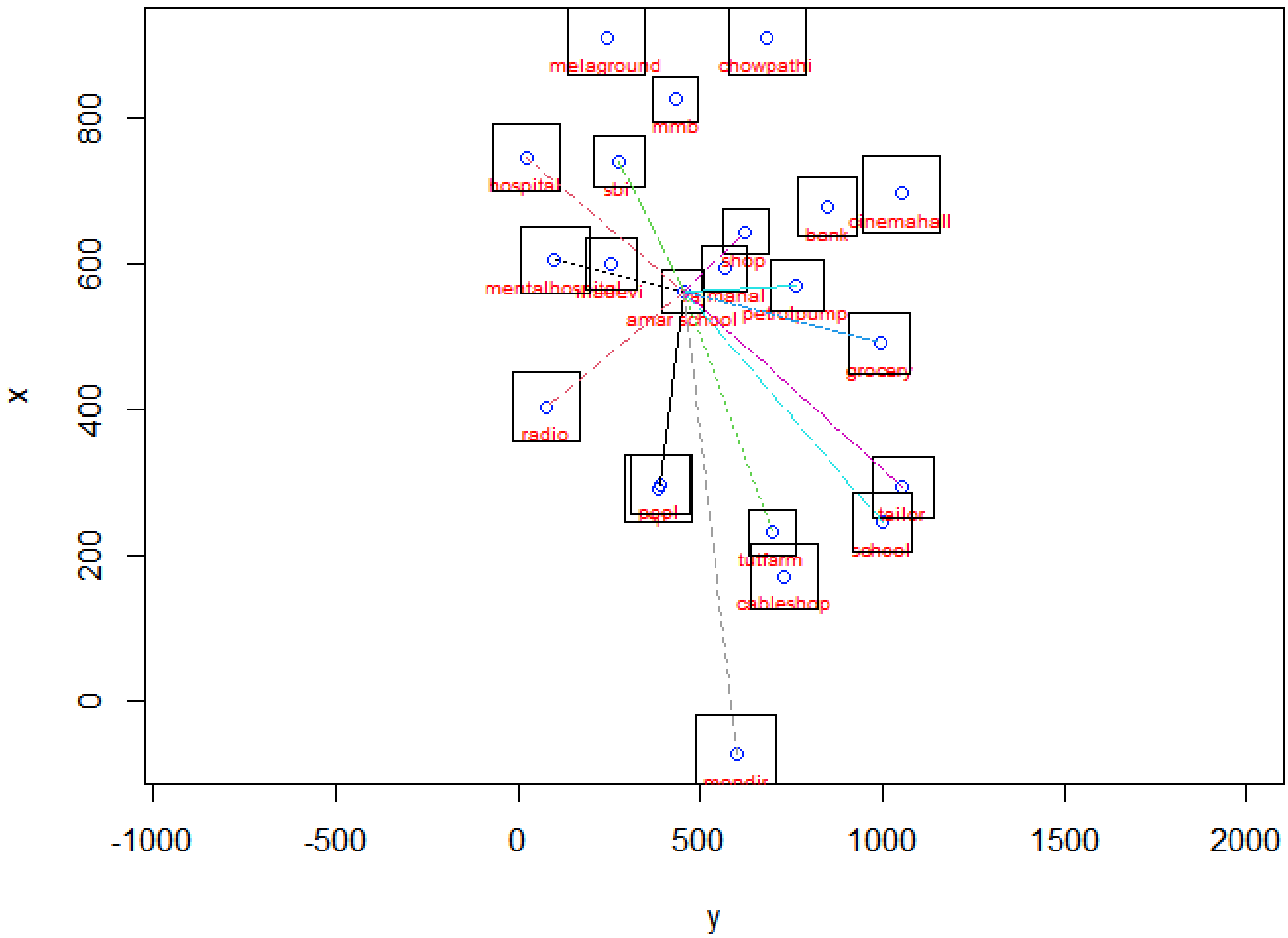
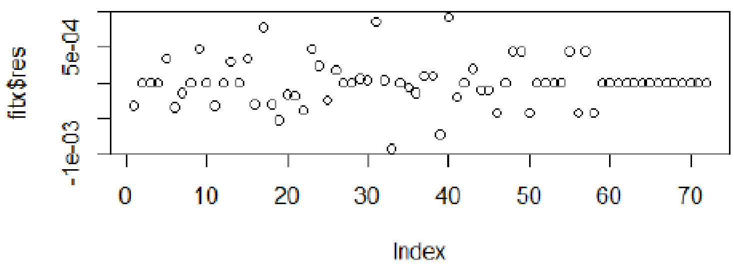


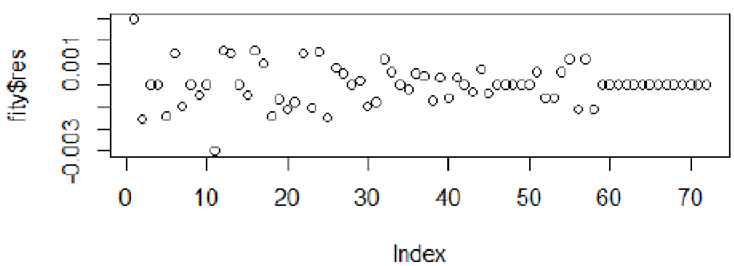
Figure 1: Map with the roads (same resolution)

as a reference. I fitted both these models in R and the coefficients of Name in both these fits correspond to the relative positions that are required. I fixed "Tufanganj Nripendra Narayan Memorial High School"(I named that as "amar school") as the reference point which is the entrance to my home. I plotted and marked all the points and connected them with a known road network. The final map is shown below:

4 Diagnostics

Now we examine the fit. First, I looked at the summary of the plots and found out the basic summary measures for both fits. The R-squared values for both fits were reported as 1, possibly due to lack of precision and the model having true R-squared values very close to 1. The standard error values were reported as 0.0006744(x-coord fit) and 0.001915(y-coord fit). These values are very low and indicate that we didn't make many errors while marking the points.Next the residual plots were checked:





The residual plots are scattered around 0 as we expect. The large number of residuals with value 0 towards the end is because the final 4-5 screenshots had very few numbers of points and I couldn't mark more than one overlapping point in each screenshot. The outliers we see in the residual plots were examined and the error is higher than others because the same point is common to a lot of screenshots which increased the probability of errors while marking. I also looked at the qqplots, the leverage plots and the influence measures. The influential points were examined again, and the error was found to be because of the higher frequency of the points itself and not any other factor. Since all the diagnostic measures seen till now indicated a very good fit, I decided not to examine the fit more. All these plots are included in the appendix.

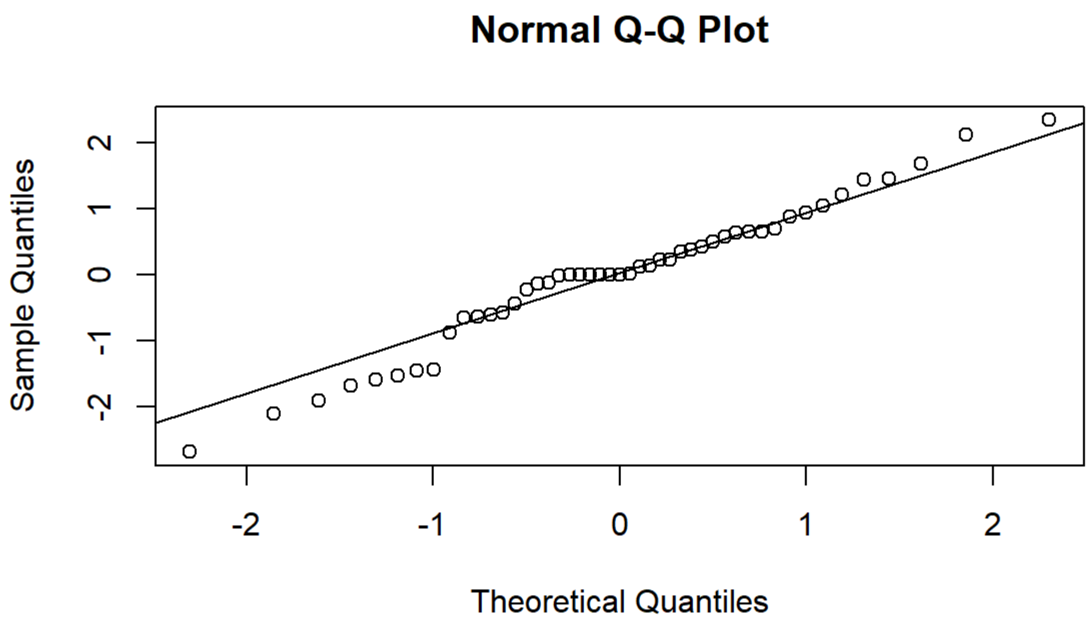


Figure 2: QQ_plot

We observe that the plot is roughly a straight line, and hence the normality assumption of the residuals is a reasonable assumption to make in this scenario

5 Incorporating Zoom Feature

I also tried to consider the case where screenshots of different(known) resolutions are allowed. For this I considered 4 new screenshots of different resolutions. The area covered by these screenshots is the left half of the area covered by our initial set of screenshots. Points were marked like before, but we also added another input which is the zoom level(resolution). Each picture has a specific zoom level which is common to all the points in the picture. The model I used for implementing this is:

$$Zoom_j * x_{ij} = Name_i + Pic_j + \epsilon$$

This model was used since zoom level corresponds to how much the image is blown up and this interpretation is captured by this model. All the other steps including diagnostics was done like before and I verified that the fit didn't have any unexplained errors. The final map obtained is shown below. Note that the relative positions of the places are same as the previous map, but the scaling factor is different.

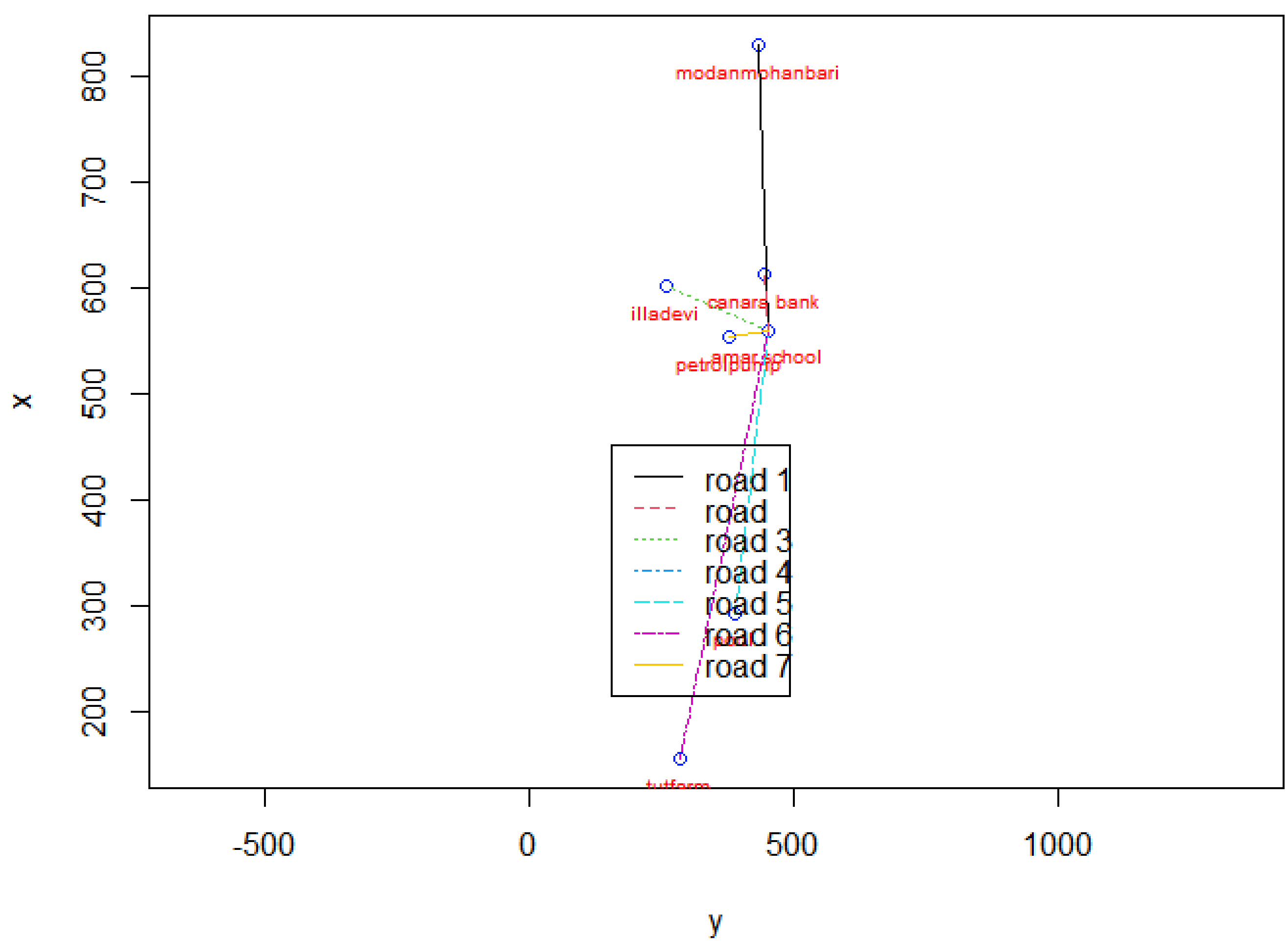


Figure 3: map sticthing (different resolution)

6 Conclusion

I was able to create a bitmap image of the chosen location using Satellite images. Since I've lived in the area for 10 years and know the location well, I was also able to manually verify that the map that we got does indeed resemble the actual lay of the land. The project can be termed as a success since we were able to create an accurate map. For further project ideas, we can try to create a map also considering rotated satellite images with different angles. This is an interesting extension to this project.

7 R code

```
   library(png)


# first we create a function that takes the places
# then we fit the model to the places,
# finally  we fit the model and join the places

# the code shall work upon typing map_stitiching(screenshots()) and entering the required details
#ABOUT map_screenshots function:
#  This function has been defined and used to read the coordinates of the locations in the screenshots,
#and store their coordinates in a data frame, and the given data frame is returned from the function.
#This returned data frame is then used for the next function makemap(), which shall make
#considerable use of this dataframe.

#ABOUT map_stitiching function:
 # This function, as earlier stated, takes the dataframe of the screenshots, the points, and their relative
#coordinates in the screenshots, respectively. The function shall use the dataframe to fit the
#linear model and plot the predicted values after joining them with lines to depict paths as and when
#specified by the user.
map_scrrenshots = function(){
  ss_no = 1
  entries = data.frame()
```

```
  while(ss_no>0){
    choice = -1;
    picture = readPNG(paste("C:/Users/SOUNAK/Desktop/screenshots/ss",ss_no,".png", sep = ""))
    d = dim(picture)[1:2]
    plot(NULL, xlim=c(0,d[2]), ylim=c(0,d[1]), ty='n', xlab="x", ylab="y", asp=1)
    rasterImage(as.raster(picture),0,0,d[2],d[1])
    if(choice != 1) {zoom = readline("What is the resolution of this image?");}
    p = locator(1)
    name = readline("What is the name of this point? ")
    entries = rbind(entries,c(ss_no,name,zoom,p$x,p$y,d[2],d[1]),stringsAsFactors=F)
    choice = as.integer(readline("Do you want to change the screenshot? \n Press 1 to stay in same screenshot \n
Press 2 to go to previous screenshot\n Press 3 to go to next screenshot \n Press 0 to stop\n "))
    if(choice == 3) ss_no = ss_no+1
    if(choice == 2) ss_no = ss_no-1
    if(choice == 0) ss_no = 0
  }
  names(entries) = c("ss", "point", "zoom", "x", "y","xlim","ylim")
  return(entries)
}

map_stitiching<-function(entries){
  entries$ss=factor(entries$ss)
  entries$point=factor(entries$point)

  # now the data variables are scaled accordingly and the predicted variables need to be multiplied with the respec
  fitx1 = lm(x~point+ss-1, entries)
  mm1 = model.matrix(fitx1);
  fity1 = lm(y~point+ss-1, entries)
  mm2 = model.matrix(fity1);
  for(i in 1 : length(entries$ss)){
    mm1[i, ] = mm1[i, ] * as.integer(entries$zoom[i]);
    mm2[i, ] = mm2[i, ] * as.integer(entries$zoom[i]);
  }

  x1 = entries$x;
```

```
y1 = entries$y;
fitx = lm(x1 ~ mm1 - 1);
fity = lm(y1 ~ mm2 - 1);
p = length(unique(entries$point))
s = length(unique(entries$ss))
finalx = fitx$coef[1:p] ;
errorx = summary(fitx)$coef[1:p,2]
finaly = fity$coef[1:p];
summary(fitx);
summary(fity);
errory = summary(fity)$coef[1:p,2]
plot(finaly, finalx, xlab="y", ylab="x", col="#001eff", asp=2)
text(finaly, finalx, sort(unique(entries$point)), cex=0.6, pos=1, col="red")
rect(finaly-errory,finalx-errorx,finaly+errory,finalx+errorx)
ss_no=1
c=0
roads=c()
t=0
while(ss_no!=0)
{
  t=as.integer(readline("Do you want to add paths? \nPress 1 if yes, 0 if no "))

  if(t==1)
  {
    name1=readline("Give the name of first place")
    name2=readline("Give the name of second place")
    z=as.integer(readline("Is it a new road? \nPress 1 if yes, 0 if no "))
    if(z==1)
    {
      c=c+1
      name3=readline("Give the name of road ")
      roads[c]=name3
    }
    segments(finaly[paste("mm2point",name1,sep="")],finalx[paste("mm1point",name1,sep="")],
    finaly[paste("mm2point",name2,sep="")],finalx[paste("mm1point",name2,sep="")],col=c,lty=c)
```

```
    }
    ss_no=t
  }
  if(c>0)
  {
    legend(min(finalx),max(finaly),roads,col=1:c,lty=1:c,bg="transparent")
  }
  scalex = abs((fitx$residuals)/as.numeric(entries$xlim))
  scaley = abs((fity$residuals)/as.numeric(entries$ylim))
  if(fitx$rank < (p+s-1)) print("Clicks given by the user do not form a connected map.
  The model fitted is not correct")
  if(max(scalex)>0.2 | max(scaley)>0.2) print(paste("Clicks are faulty. The fitted map is not correct."))

  #model diagnostics
  resx=resid(fitx)
  plot(fitted(fitx),resx)
  qqnorm(resx)
  qqline(resx)


}

map_stitiching(map_scrrenshots())
```