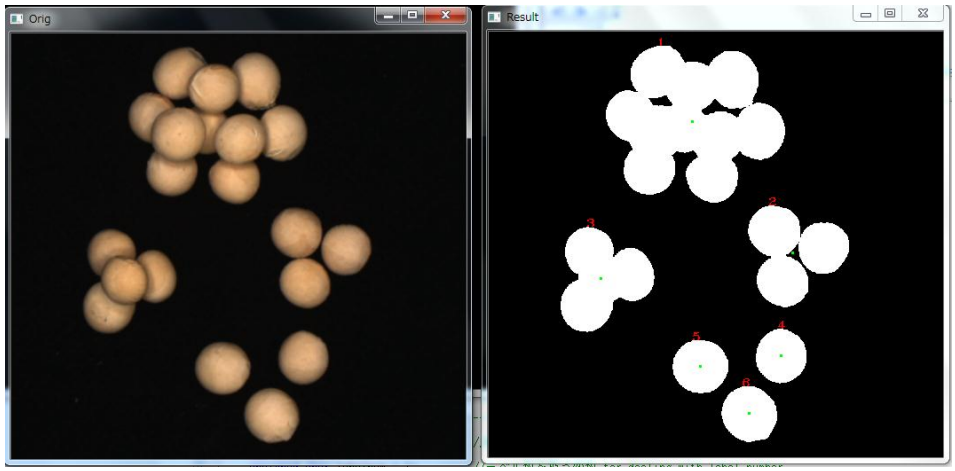
Outline of sample program

1. Read 4 sample images (test image to construct algorithm) one after another.
2. Split RGB image, and binarize R image by using 40 value as threshold.
3. Process labeling
4. Calculate gravity position
5. Show label value & gravity position on binary image.
   (Red number indicates label value, green point indicates gravity position)



6. Save result image and text files to folder (C:¥Shiigi)

Details

1. Read 4 sample images (test image to construct algorithm) one after another.
   Here, it is important to handle character string for reading image one after another.

```
//------画像の前名前と拡張子を変数に書き込む------------------------------//
//------write pre-name and extension of input image to variables ---------//
    strcpy(name1,"Sample");
    strcpy(name2,".bmp");

//------ファイルオープン処理------------------------------//
//------proccess to open file ---------//

    if((fp=fopen("C:¥¥Shiigi¥¥Result.txt","w")) == NULL){
        printf("file open error!!¥n");
        exit(1);
    }

//------画像を順次読み込むためのforループ----------------------------------//
//------"for roop" for sequential reading image----------------------------//
    for(i=0;i<4;i++){

//------画像を順次読み込むための名前変更----------------------------------//
//------change name for sequential reading image----------------------------//
        strcpy(in_name, "");
        sprintf(num,"%d",i);
        strcat(in_name, name1);
        strcat(in_name, num);
        strcat(in_name, name2);

//------画像の読み込み------------------------------------------------------//
//------read image----------------------------------------------------------//

        IplImage *Orig_image = cvLoadImage( in_name, CV_LOAD_IMAGE_ANYDEPTH | CV_LOAD_IMAGE_ANYCOLOR );
```

"i" in for roop is used as also number] of image. "strcpy" copy character string. "sprintf" change from integer to character. "strcat" connect character string.

Flow of process:

1) "Sample" is copied to name1.
2) ".bmp" is copied to name2.
3) "in_name" is initialized.
4) "i" is input to "num" as character.
5) Connect "in_name" to "name1" → "in_name" is Sample
6) Connect "in_name" to "num" → "in_name" is "Sample0" (when "i" is 0)
7) Connect "in_name" to "name2" → "in_name" is "Sample0.bmp"

When 4 of "for(i=0;i<4;i++)" change to 20, images from "Sample0.bmp" to "Sample19.bmp" can be read.

"cvLoadImage" function (OpenCV function)

This function read image.

First argument: name of image, Second argument: set image depth & color

Return: pointer of "IplImage" structure

2. Split RGB image and binarize R image using threshold value of 40

```
//------画像の生成------------------------------------------------------------//
//------create image---------------------------------------------------------//

        IplImage *R_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 1 );           //Red
        IplImage *G_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 1 );           //Green
        IplImage *B_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 1 );           //Blue
        IplImage *Binary_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 1 );      //Binary
        IplImage *FinalLabel_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 1 );  //Label
        IplImage *Center_Cal_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 1 );  //Center
        IplImage *Result_image = cvCreateImage ( cvGetSize(Orig_image), IPL_DEPTH_8U, 3 );      //Result

//------画像の分離------------------------------------------------------------//
//------split color image to RGB ---------------------------------------------//

        cvSplit( Orig_image, B_image, G_image, R_image, NULL );

//------2値化-----------------------------------------------------------------//
//------binarization ----------------------------------------------------------//

        for( y=0; y < Orig_image->height; y++){
            for ( x=0; x < Orig_image->width; x++){

                p_image = R_image->imageData[y * R_image->width + x];

                //2値化の閾値を40以上に設定
                //set more than 40 as threshhold value for binarization
                if(p_image>=40){
                    Binary_image->imageData[y * Binary_image->width + x] = (unsigned char) 255;
                }else{
                    Binary_image->imageData[y * Binary_image->width + x] = (unsigned char) 0;
                }
            }
        }
```

“cvCreateImage” function

This function create image.

First argument: size of image (return image size by using “cvGetSize” function)

Second argument: image depth (IPL_DEPTH_8U indicates unsigned 8 bit)

Third argument: number of channels (1 is gray image, 3 is color image)

Return: pointer of “IplImage” structure

“cvSplit” function

This function split RGB image to R, G, B images.

First argument: “IplImage” structure of color image

Second argument: “IplImage” structure of B image

Third argument: “IplImage” structure of G image

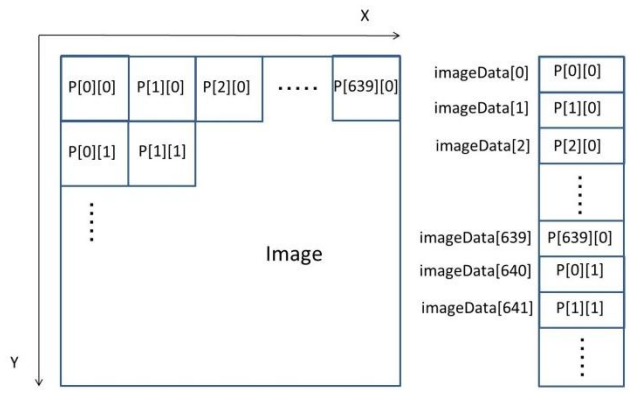
Forth argument: “IplImage” structure of R image

In for roop:

Orig_image->height is member of “IplImage” structure indicates height of image

Orig_image->width is member of “IplImage” structure indicates width of image

R_image->imageData[] is member of “IplImage” structure indicates array of value of image



Flow of process for binarization

1) Input “p_image” to data of R image
2) If “p_image” is more than 40
   Value of “Binary_image” is set 255

If “p_image” is less than 40

Value of “Binary_image” is set 0

3) Labeling

```
この先は最終的に2値化した結果をラベリング，重心位置計算，結果画像の保存＆表示，結果のテキスト出力に用いると便利です----//

最後に2値化した画像をラベリング用の画像にコピー-----------------------//
cvCopy(Binary_image, FinalLabel_image, NULL);
cvZero(Center_Cal_image);

ラベリング----------------------------------------------------------//
labeling ----------------------------------------------------------//

labelnum=1;
for(y=0;y<FinalLabel_image->height;y++){
    for(x=0;x<FinalLabel_image->width;x++){

        p_binari = FinalLabel_image->imageData[y * FinalLabel_image->width + x];
        if(p_binari == 255){
            //ラベリング関数
            //labeling function
            cvFloodFill( FinalLabel_image, cvPoint(x,y), cvScalarAll(labelnum), cvScalarAll(0), cvScalarAll(0), &comp[labelnum], 4, NULL );

            //面積50以下は除去
            //remove if are is less than 50
            if(comp[labelnum].area <= 50){
                cvFloodFill( FinalLabel_image, cvPoint(x,y), cvScalarAll(0), cvScalarAll(0), cvScalarAll(0), &comp[0], 4, NULL );
            }else{
                //もしラベル数が255以上になったとき画像上で表現できないためエラーとして扱う
                //deal with error when label number more than 255 because label number can not show on image
                if(labelnum == 255){
                    printf("limit¥n");
                    exit(1);
                }
                //show on command prompt
                //printf("labelnum = %d, area = %lf¥n",labelnum, comp[labelnum].area);
                labelnum ++;
            }
        }
    }
}
```

“cvFloodFill” function

This function fills in connected area.

First argument: image (“IplImage” structure)

Second argument: position for starting connection

Third argument: value for filling

Forth argument: condition to connect pixel (minimum allowance)

Fifth argument: condition to connect pixel (maximum allowance)

Sixth argument: “CvConnectedComp” structure (information of filled area such as size..)

Seventh argument: connectedness

Eighth argument: mask

Flow of process

1) “Binary_Image” is copied to “FinalLavel_image”
2) “labelnum” is initialized. (Value is 1)
3) To “for roop”
4) To find position of 255 value on “FinalLavel_image” from upper left
5) If value is 255

   5)-1. labeling using “cvFloodFill” function

Position for starting connection: found position of 255

Value for filling: “labelnum”

5)-2. noise reduction

To remove area when size of this area is less than 50

5)-3. renewal of label value (“labelnum”)

To add 1 to “labelnum” when size of are is more than 50

This program can not process when “labelnum” is more than 255 because this labeling process is conducted on 8 bit image.

4. Calculate gravity position of each labeled area

   Gravity position is obtained by using “cvMoments” & “cvGetSpatialMoment” functions

“cvMoments” function

This function calculates image moment within 3 orders and saves result at Second argument.

First argument: image (“IplImage” structure)

Second argument: pointer for structure of image moments

Third argument: flag

“cvGetSpatialMoment” function

This function calculates spatial moment by image moment as below.

$M_{x\_order,\ y\_order} = Sum_{x,\ y}(I(x,y) \cdot x^{x\_order} \cdot y^{y\_order})$

Where, $I(x,y)$ is intensity of (x,y).

Flow of process

1) To create binary image (“Cenetr_Cal_image”) by using each label value
2) To obtain image moments (“CvMoments moments”) by using “cvMoments” function
3) To obtain 0 order moment (area size: m_00), 1 order (x) & 0 order (y) moment (m_10), 0 order (x) & 1 order (y) moment (m_01) by using “cvGetSpatialMoment” function
4) To calculate gravity position (Gravity[label value][x or y (x: 0, y: 1)])
5) To initialize “Center_Cal_image” by using “cvZero” function

```
//----------重心位置計算------------------------------------------------------//
//----------Calculate position of gravity-------------------------------------//
        fprintf(fp, "Lable, X, Y¥n");
        for(j=1;j<labelnum;j++){
            for(y=0;y<FinalLabel_image->height;y++){
                for(x=0;x<FinalLabel_image->width;x++){
                    p_binari = FinalLabel_image->imageData[y * FinalLabel_image->width + x];
                    if(p_binari==j){
                        Center_Cal_image->imageData[y * FinalLabel_image->width + x] = (unsigned char) 255;
                    }
                }
            }

            cvMoments(Center_Cal_image,&moments,0);
            m_00 = cvGetSpatialMoment(&moments,0,0);
            m_10 = cvGetSpatialMoment(&moments,1,0);
            m_01 = cvGetSpatialMoment(&moments,0,1);
            Gravity[j][0] = (int)(m_10/m_00);
            Gravity[j][1] = (int)(m_01/m_00);

            cvZero(Center_Cal_image);

            fprintf(fp, "%d, %d, %d¥n", j, Gravity[j][0], Gravity[j][1]);
        }

        fprintf(fp, "¥n");
```

7. Show label value & gravity position on binary image.
   (Red number indicates label value, green point indicates gravity position)

Show label value

“cvInitFont” function

This function initializes structure for font to send “cvPutText” function

First argument: pointer of structure for font (CvFont　font[0])

Second argument: identifier of font

Third argument: rate of width, fourth argument: rate of hight

“cvPutText” function

Write character string with designated color & font on image

First argument: image (“IplImage” structure)

Second argument: character string for writing on image

Third argument: position of upper left of first character

Fourth argument: pointer of structure of font

Fifth argument: color of character

Flow of process

1) To set font by using “cvInitFont” function
2) To initialize label value (“labelnumcounter”)
3) If pixel with label value is found
   3-1. to change label value (“labelnumcounter”) to character string (“text”) by using “sprintf” function
   3-2. to write character string of “text” with “font[0]” & “color” at position of x, y on “Result_image” by using “cvPutText” function
   Color information is defined by “CvScalar” structure.
   CvScalar color = {Blue, Green, Red, 0}
4) To create binary image from labeled image

```
//------ラベル値の表示----------------------------------------------------------//
//------show label number ------------------------------------------------------//

        //フォントの設定
        //set font
        cvInitFont(&font[0], CV_FONT_HERSHEY_COMPLEX_SMALL, 0.7,0.7);

        labelnumcounter=1;
        for(y=0;y<FinalLabel_image->height;y++){
            for(x=0;x<FinalLabel_image->width;x++){

                p_binari = FinalLabel_image->imageData[y * FinalLabel_image->width + x];
                if(p_binari > 0){
                    if(labelnumcounter==p_binari){
                        sprintf(text, "%d", labelnumcounter);
                        //ラベル数の書き込み
                        //write label number
                        cvPutText (Result_image, text, cvPoint (x,y), &font[0], color);
                        labelnumcounter++;
                    }
                    //2値化
                    //binarization
                    Result_image->imageData[(y * FinalLabel_image->width + x)*3] = (unsigned char) 255;
                    Result_image->imageData[(y * FinalLabel_image->width + x)*3 + 1] = (unsigned char) 255;
                    Result_image->imageData[(y * FinalLabel_image->width + x)*3 + 2] = (unsigned char) 255;
                }else{
                    Result_image->imageData[(y * FinalLabel_image->width + x)*3] = (unsigned char) 0;
                    Result_image->imageData[(y * FinalLabel_image->width + x)*3 + 1] = (unsigned char) 0;
                    Result_image->imageData[(y * FinalLabel_image->width + x)*3 + 2] = (unsigned char) 0;
                }
            }
        }
```

Show gravity position

To show gravity position using 3 x 3 area with green color

```
-----重心位置表示-------------------------------------------------------------//
-----show position of gravity-------------------------------------------------//
 for(j=1;j<labelnum;j++){
     for(y=-1;y<=1;y++){
         for(x=-1;x<=1;x++){
             Result_image->imageData[((Gravity[j][1]+y) * FinalLabel_image->width + (Gravity[j][0]+x))*3] = (unsigned char) 0;
             Result_image->imageData[((Gravity[j][1]+y) * FinalLabel_image->width + (Gravity[j][0]+x))*3 + 1] = (unsigned char) 255;
             Result_image->imageData[((Gravity[j][1]+y) * FinalLabel_image->width + (Gravity[j][0]+x))*3 + 2] = (unsigned char) 0;
         }
     }
 }
```

6) Save result image and text files to folder (C:¥Shiigi)

To write result on text file

1. To write image name & label value (number of object) before gravity position calculation
2. To write label value, x position of gravity and y position of gravity

```
fprintf(fp, "Image file name, number of sample¥n");
fprintf(fp, "%s, %d¥n",in_name,labelnumcounter-1);

//-----重心位置計算-------------------------------------------------------------//
//-----Calculate position of gravity--------------------------------------------//
fprintf(fp, "Lable, X, Y¥n");
for(j=1;j<labelnum;j++){
    for(y=0;y<FinalLabel_image->height;y++){
        for(x=0;x<FinalLabel_image->width;x++){
            p_binari = FinalLabel_image->imageData[y * FinalLabel_image->width + x];
            if(p_binari==j){
                Center_Cal_image->imageData[y * FinalLabel_image->width + x] = (unsigned char) 255;
            }
        }
    }

    cvMoments(Center_Cal_image,&moments,0);
    m_00 = cvGetSpatialMoment(&moments,0,0);
    m_10 = cvGetSpatialMoment(&moments,1,0);
    m_01 = cvGetSpatialMoment(&moments,0,1);
    Gravity[j][0] = (int)(m_10/m_00);
    Gravity[j][1] = (int)(m_01/m_00);

    cvZero(Center_Cal_image);

    fprintf(fp, "%d, %d, %d¥n", j, Gravity[j][0], Gravity[j][1]);
}

fprintf(fp, "¥n");
```

To save image & text file

Show image

"cvNamedWindow" function

This function create window

First argument: name of window, Second argument: size of window

"cvShowImage" function

This function show image on designed window

First argument: name of window, Second argument: image ("IplImage" structure)

"cvWaitKey" function

This function stop program temporary

First argument: delay time (msec)　If this value is 0, program wait by input key

“cvDestroyWindow” function

This function destroy window

First argument: window name

Save image

“cvSaveImage” function

This function save image

First argument: name of file

Second argument: image (“IplImage” structure)

Release image

“cvReleaseImage” function

This function release memory of image

First argument: image(“IplImage” structure)

```
//window の作成
//create window
cvNamedWindow( "Orig", CV_WINDOW_AUTOSIZE );
cvNamedWindow( "Result", CV_WINDOW_AUTOSIZE );

//画像の表示
//show image
cvShowImage( "Orig", Orig_image );
cvShowImage( "Result", Result_image );

//wait input key
cvWaitKey(0);

//画像の保存
//save image
strcpy(out_name,"C:¥¥Shiigi¥¥Result");
strcat(out_name, num);
strcat(out_name, name2);
cvSaveImage(out_name, Result_image );

//windowの破棄
//destroy window
cvDestroyWindow( "Orig" );
cvDestroyWindow( "Result" );

//メモリーの解放
//release memory
cvReleaseImage( &Orig_image );
cvReleaseImage( &R_image );
cvReleaseImage( &G_image );
cvReleaseImage( &B_image );
cvReleaseImage( &Binary_image );
cvReleaseImage( &FinalLabel_image );
cvReleaseImage( &Center_Cal_image );
cvReleaseImage( &Result_image );
```