フィルター処理　静止画プログラム例

```
PImage img;

void setup() {
  size(640, 480);
  img = loadImage("capImg1.png");
}

void draw() {
  image(img, 0, 0);
  filter(THRESHOLD, 0.5);
  //filter(GRAY);
  //filter(INVERT);
  //filter(POSTERIZE, 4);
  //filter(BLUR, 6);
}
```

フィルター処理　動画プログラム例

```
import processing.video.*;

Movie movie;

void setup() {
  size(960, 540);
  movie = new Movie(this, "face_movie.mov");
  movie.loop();
}

void movieEvent(Movie movie) {
  movie.read();
}

void draw() {
  image(movie, 0, 0, width, height);
  filter(THRESHOLD, 0.5);
  //filter(GRAY);
  //filter(INVERT);
  //filter(POSTERIZE, 4);
  //filter(BLUR, 6);
}
```

フィルター処理　カメラ映像プログラム例

```
import processing.video.*;
Capture capImg;

void setup() {
  size(640, 480);
  String[] cameras = Capture.list();
  println("Available cameras:");
  for (int i = 0; i < cameras.length; i++) {
    println(i, cameras[i]);
  }
  capImg = new Capture(this, width, height, cameras[0]);
  //capImg = new Capture(this, width, height, cameras[3]);  // for Surface PC
  capImg.start();
}

void captureEvent(Capture capImg) {
  capImg.read();
}

void draw() {
  image(capImg, 0, 0);
  filter(THRESHOLD, 0.5);
  //filter(GRAY);
  //filter(INVERT);
  //filter(POSTERIZE, 4);
  //filter(BLUR, 6);
}
```

背景差分　静止画プログラム例

```
PImage bgImg, capImg;

void setup() {
  size(640, 480);
  bgImg = loadImage("capImg1.png");
  capImg = loadImage("capImg2.png");
}

void draw() {
  image(bgImg, 0, 0);
  blend(capImg, 0, 0, width, height, 0, 0, width, height, DIFFERENCE);
}
```

背景差分　動画プログラム例

```
import processing.video.*;

Movie movie;

PImage bgImg;
boolean bgFlg = false;

int count=0;

void setup() {
  size(960, 540);
  movie = new Movie(this, "face_movie.mov");
  movie.loop();
}

void movieEvent(Movie movie) {
  movie.read();
  if (count>10) {
    if (bgFlg == false) {
      bgImg = movie.get(0, 0, width, height);
      bgFlg = true;
    }
  }
  count++;
}
```

```
void draw() {
  if (bgImg != null) {
    image(bgImg, 0, 0);
    blend(movie, 0, 0, width, height, 0, 0, width, height, DIFFERENCE);
  }
}

void keyPressed() {
  if (key == ' ') {
    bgFlg = false;
  }
}
```

背景差分　カメラ映像プログラム例

```
import processing.video.*;
Capture capImg;

PImage bgImg;
boolean bgFlg = false;

int count=0;

void setup() {
  size(640, 480);
  String[] cameras = Capture.list();
  println("Available cameras:");
  for (int i = 0; i < cameras.length; i++) {
    println(i, cameras[i]);
  }
  capImg = new Capture(this, width, height, cameras[0]);
  //capImg = new Capture(this, width, height, cameras[3]);  // for Surface PC
  capImg.start();
}
```

```
void captureEvent(Capture capImg) {
  capImg.read();
  if (count>10) {
    if (bgFlg == false) {
      bgImg = capImg.get(0, 0, width, height);
      bgFlg = true;
    }
  }
  count++;
}

void draw() {
  if (bgImg != null) {
    image(bgImg, 0, 0);
    blend(capImg, 0, 0, width, height, 0, 0, width, height,
DIFFERENCE);
  }
}

void keyPressed() {
  if (key == ' ') {
    bgFlg = false;
  }
}
```

顔検出　静止画プログラム例

```
import gab.opencv.*;
import java.awt.Rectangle;
OpenCV opencv;
Rectangle[] faces;

void setup() {
  opencv = new OpenCV(this, "test.jpg");
  size(640, 640);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  faces = opencv.detect();
}

void draw() {
  image(opencv.getInput(), 0, 0);
  noFill();
  stroke(0, 255, 0);
  strokeWeight(3);
  for (int i = 0; i < faces.length; i++) {
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}
```

顔検出　動画プログラム例

```
import gab.opencv.*;
import java.awt.*;
import processing.video.*;

Movie movie;
OpenCV opencv;
PImage img;

void setup() {
  size(960, 540);
  movie = new Movie(this, "face_movie.mov");
  movie.loop();
  frameRate(30);
  opencv = new OpenCV(this, width, height);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  strokeWeight(5);
  stroke(255, 0, 0);
  noFill();
  img = loadImage("face.png");
}
```

```
void movieEvent(Movie movie) {
  movie.read();
}

void draw() {
  opencv.loadImage(movie);
  image(movie, 0, 0);
  Rectangle[] faces = opencv.detect();
  for (int i = 0; i < faces.length; i++) {
    image(img, faces[i].x, faces[i].y, faces[i].width, faces[i].height );
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}
```

顔検出　カメラ映像プログラム例

```
import gab.opencv.*;
import java.awt.*;
import processing.video.*;
Capture capImg;
OpenCV opencv;
PImage img;

void setup() {
  size(640, 480);
  //size(960, 540);  // for Mac
  String[] cameras = Capture.list();
  println("Available cameras:");
  for (int i = 0; i < cameras.length; i++) {
    println(i, cameras[i]);
  }
  capImg = new Capture(this, width, height, cameras[0]);
  //capImg = new Capture(this, width, height, cameras[3]);  // for Surface PC
  capImg.start();
  opencv = new OpenCV(this, width, height);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  strokeWeight(5);
  stroke(255, 0, 0);
  noFill();
  img = loadImage("face.png");
}
```

```
void captureEvent(Capture capImg) {
  capImg.read();
}

void draw() {
  opencv.loadImage(capImg);
  image(capImg, 0, 0);
  Rectangle[] faces = opencv.detect();
  for (int i = 0; i < faces.length; i++) {
    if (i == 0) {
      image(img, faces[i].x, faces[i].y, faces[i].width,
faces[i].height );
    } else {
      rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
    }
  }
}
```