

Notas sobre vídeo:

- OpenCV grava/lê vídeos usando a biblioteca de codecs FFmpeg.
- Sugestão de programa para visualizar vídeos gerados por OpenCV:
 - o Windows/Linux: [smplyer](#)
 - o iOS: [movie player](#)
- Sugestão de programas em Windows para editar vídeo e visualizar vídeo quadro-a-quadro:
 - o [VIRTUALDUB](#)
 - o [VIRTUALDUBMOD](#).
- Sugestão de CODECs de vídeo para Windows: [FFDSHOW](#). Se você não instalar este programa, “Windows Media Player” e similares não conseguirão mostrar corretamente alguns tipos de vídeos.
- A versão `ffdshow_rev3572_20100913_clsids.exe` parece ser a última que traz todos os codificadores. Aparentemente, as versões mais novas trazem poucos codificadores.
- No OpenCV, escrita de vídeos em níveis de cinza às vezes dá erros inexplicáveis. Isto é, abrindo `VideoWriter` com `isColor=false`, muitas vezes OpenCV gera um vídeo sem quadros:

```
VideoWriter::VideoWriter(const string& filename, int fourcc, double fps, Size frame-  
Size, bool isColor=true)
```

Recomendo usar `VideoWriter` somente com `isColor=true`.

1) Gera os quadros de um vídeo e grava-os.

```
//criavideo.cpp
//pos-2014
#include <cekeikon.h>
int main() {
    VideoWriter vo("criavideo.avi",
                  CV_FOURCC('X','V','I','D'),
                  30,
                  Size(320,240));
    Mat_<COR> a(240,320);
    for (int q=0; q<200; q++) {
        for (int l=0; l<a.rows; l++)
            for (int c=0; c<a.cols; c++)
                a(l,c)=COR((2*q)%256,(l+q)%256,(c+q)%256);
        vo << a;
    }
}
```

Nota: **CV_FOURCC('X','V','I','D')** fará o programa gravar um vídeo usando codificador MPEG4 (extensão .AVI).

O vídeo compactado usando esta opção tipicamente fica umas 20 vezes menor do que a versão sem compactação.

Nota: **CV_FOURCC('P','I','M','1')** fará o programa gravar um vídeo usando codificador MPEG1 (extensão .MPG).

Nota: Se trocar **CV_FOURCC('X','V','I','D')** por 0, o programa vai gerar vídeo descompactado (extensão .AVI) [só em Windows?]

Nota: Se trocar **CV_FOURCC('X','V','I','D')** por -1, o programa vai perguntar ao usuário o codificador a ser usado [só em Windows?]

Nota: O programa vai gravar o vídeo com 30 quadros por segundo.

2) Le um vídeo, modifica-o, e grava o vídeo modificado.

```
//leimp.cpp: Le e grava video.
//pos-2014
#include <cekeikon.h>
int main()
{ VideoCapture vi("criavideo.avi");
  if (!vi.isOpened()) erro("Erro abertura video entrada");
  VideoWriter vo("leimp.avi",
                -1, //Isto nao funciona em Linux
                30, Size(320,240));
  if (!vo.isOpened()) erro("Erro abertura video saida");
  Mat_<COR> a;
  while (true) {
    vi >> a;
    if (!a.data) break;
    for (unsigned i=0; i<a.total(); i++)
      a(i)[2]=255;
    vo << a;
  }
}
```

Nota: A codec opção -1 só funciona em Windows. Use CV_FOURCC('X','V','I','D') em Linux, com extensão .AVI).

```
//leimp.cpp:
//Le e grava video,
//fazendo tamanho e fps de saida iguais `as de entrada
#include <cekeikon.h>
int main()
{ VideoCapture vi("criavideo.avi");
  if (!vi.isOpened()) erro("Erro abertura video entrada");
  int nc=vi.get(CV_CAP_PROP_FRAME_WIDTH);
  int nl=vi.get(CV_CAP_PROP_FRAME_HEIGHT);
  int fps=vi.get(CV_CAP_PROP_FPS);
  VideoWriter vo("teste.avi",
                CV_FOURCC('X','V','I','D'),
                fps, Size(nc,nl));
  if (!vo.isOpened()) erro("Erro abertura video saida");
  Mat_<COR> a;
  while (true) {
    vi >> a;
    if (!a.data) break;
    for (unsigned i=0; i<a.total(); i++)
      a(i)[2]=255;
    vo << a;
  }
}
```

Possíveis codecs (tem que testar quais funcionam):

All systems:

- CV_FOURCC('P','I','M','1') = MPEG-1 codec (extensão .MPG)
- CV_FOURCC('M','J','P','G') = motion-jpeg codec (OK)
- CV_FOURCC('M','P','4','2') = MPEG-4.2 codec (OK)
- CV_FOURCC('D','I','V','3') = MPEG-4.3 codec (OK)
- CV_FOURCC('D','I','V','X') = MPEG-4 codec (OK)
- CV_FOURCC('U','2','6','3') = H263 codec (OK)
- CV_FOURCC('I','2','6','3') = H263I codec (erro)
- CV_FOURCC('F','L','V','1') = FLV1 codec (OK)

Windows only:

- -1 = allows the user to choose the codec from a dialog at runtime
- 0 = creates an uncompressed AVI file (the filename must have a .avi extension)

Outros:

```
CV_FOURCC('H','F','Y','U') # HuffYUV (OK)
CV_FOURCC('D','R','A','C') # BBC Dirac (erro)
CV_FOURCC('X','V','I','D') # MPEG-4 Part 2 (OK - parece igual a DIVX)
CV_FOURCC('X','2','6','4') # MPEG-4 Part 10 (aka. H.264 or AVC - erro)
CV_FOURCC('M','P','1','V') # MPEG-1 video (erro)
```

		.AVI	.MP4	.MPG	.MP2
PIM1	MPEG1	ffplay		OK	
MJPG	motion-jpeg	OK			
MP42	MPEG-4.2	OK	erro		
DIV3	MPEG-4.3	OK	erro		
DIVX	MPEG-4	OK	erro	ffplay	
U263	H263	OK	erro		
I263	H263I	erro	erro		
FLV1	FLV1	OK			
-1	user - só Windows	OK			
0	uncompressed - só Windows	OK			
HFYU	HuffYUV	OK			
DRAC	BBC Dirac	erro			
XVID	MPEG4-2	OK	erro		
X264	MPEG4-H264	erro	erro		
MP1V	MPEG1	erro			
MPEG	MPEG2VIDEO	OK	erro	OK	OK

2) Captura de WEBCAM

```
// Pos-2011: Pega video do webcam e mostra na tela.
// Resolucao padrao: 480x640
// webcam.cpp
#include <cekeikon.h>
int main()
{ VideoCapture vi(0); // 0=default camera
  if (!vi.isOpened()) erro("Erro abertura webcam");
  Mat_<COR> a;
  int ch;
  namedWindow("janela");
  do {
    vi >> a;
    imshow("janela",a);
    ch=waitKey(25);
  } while (ch!=27);
}
```

```
//Pos-2011: Pega vídeo low-res do webcam e mostra na tela.
#include <cekeikon.h>
int main()
{ VideoCapture vi(0);
  vi.set(CV_CAP_PROP_FRAME_WIDTH,320);
  vi.set(CV_CAP_PROP_FRAME_HEIGHT,240);
  if (!vi.isOpened()) erro("Erro abertura webcam");
  Mat_<COR> a;
  int ch;
  namedWindow("janela");
  do {
    vi >> a;
    imshow("janela",a);
    ch=waitKey(25);
  } while (ch!=27);
}
```

```

//Pega quadros da webcam, mostra e grava (ate apertar uma tecla)
//webcam3.cpp - pos2013
#include <cekeikon.h>
int main()
{ VideoCapture w(0);
  if (!w.isOpened()) erro("Erro: Abertura de webcam");

  VideoWriter vo("webcam3_xvid.avi",CV_FOURCC('X','V','I','D'),
                30,Size(640,480));
  // XVID == MPEG4
  // 0 == AVI nao-compactado
  // -1 == pergunta ao usuario o codec a ser usado
  // 30 == fps

  Mat_<COR> a;
  namedWindow("janela",0);
  while (true) {
    w >> a;
    imshow("janela",a);
    vo << a;
    if (waitKey(30)>=0) break;
  }
}

```

```

//levideo.cpp - pos2013
//Le um vídeo e mostra na tela
#include <cekeikon.h>

int main()
{ VideoCapture vi("webcam3_1.avi");

  Mat_<COR> a;
  namedWindow("janela",0);
  while (true) {
    vi >> a;
    if (a.data==0) break;
    //
    imshow("janela",a);
    if (waitKey(30)>=0) break;
  }
}

```

```
//elimfundo.cpp - pos2016
//calcula diferenca absoluta entre o quadro atual e o anterior
#include <cekeikon.h>

int main() {
    VideoCapture vi(0); // 0=default camera
    if (!vi.isOpened()) erro("Erro abertura webcam");

    Mat_<COR> ant,atu,dif;
    vi >> atu;
    int ch;
    namedWindow("janela");
    do {
        ant=atu.clone();
        vi >> atu;
        absdiff(ant,atu,dif);
        dif = 8*dif;
        imshow("janela",dif);
        ch=waitKey(33);
    } while (ch!=27);
}
```

```

// Difquad: Captura webcam, calcula diferenca quadro a quadro
// e mostra na tela
// Grad2015 - difquad.cpp
#include <cekeikon.h>

Mat_<COR> diferenca(Mat_<COR> ant, Mat_<COR> atu) {
    if (ant.size()!=atu.size()) erro("Erro: dimensoes diferentes");
    Mat_<COR> dif(ant.size());
    for (unsigned i=0; i<dif.total(); i++) {
        dif(i)[0]=abs(ant(i)[0]-atu(i)[0]);
        dif(i)[1]=abs(ant(i)[1]-atu(i)[1]);
        dif(i)[2]=abs(ant(i)[2]-atu(i)[2]);
    }
    return dif;
}

int main() {
    VideoCapture vi(0); // 0=default camera
    if (!vi.isOpened()) erro("Erro abertura webcam");
    Mat_<COR> atu,ant,dif;
    int ch;
    namedWindow("janela");
    vi >> atu;
    do {
        ant=atu.clone();
        vi >> atu;
        dif=4*diferenca(ant,atu);
        imshow("janela",dif);
        ch=waitKey(25);
    } while (ch!=27);
}

```



```

//Subtraí fundo com conversao para grayscale e correcao de contraste
// subtraig.cpp - pos2011
#include <cekeikon.h>

double amostra(Mat_<GRY> a)
{ int soma=0;
  for (int l=5; l<18; l++)
    for (int c=5; c<15; c++)
      soma = soma + a(l,c);
  return soma / 130.0;
}

void subtraiMax(Mat_<COR> a, Mat_<COR> f, Mat_<COR>& s)
{ if (a.rows!=f.rows || a.cols!=f.cols)
  erro("Erro: Dimensoes diferentes");
  s.create(a.size());
  Mat_<GRY> ag; converte(a,ag); double ma=amostra(ag);
  Mat_<GRY> fg; converte(f,fg); double mf=amostra(fg);
  for (int l=0; l<a.rows; l++)
    for (int c=0; c<a.cols; c++) {
      int m=arredonda(abs((mf/ma)*ag(l,c)-fg(l,c)));
      if (m>64) m=255; else m=0;
      s(l,c)=COR(m,m,m);
    }
}

int main()
{ VideoCapture vi("trafego.avi");
  VideoWriter vo("subtraig.avi",
    CV_FOURCC('X','V','I','D'), // MPEG4
    15,Size(640,480)
  );
  Mat_<COR> a,f,s; le(f,"fundo.jpg");
  do {
    vi >> a;
    if (!a.data) break;
    subtraiMax(a,f,s);
    vo << s;
  } while (true);
}

```

Comando para converter um vídeo para formato AVI (sem áudio) que Proeikon e Cekeikon conseguem ler usando ffmpeg:

```
>ffmpeg -i nomevideo.ext -an nomevideo.avi
```