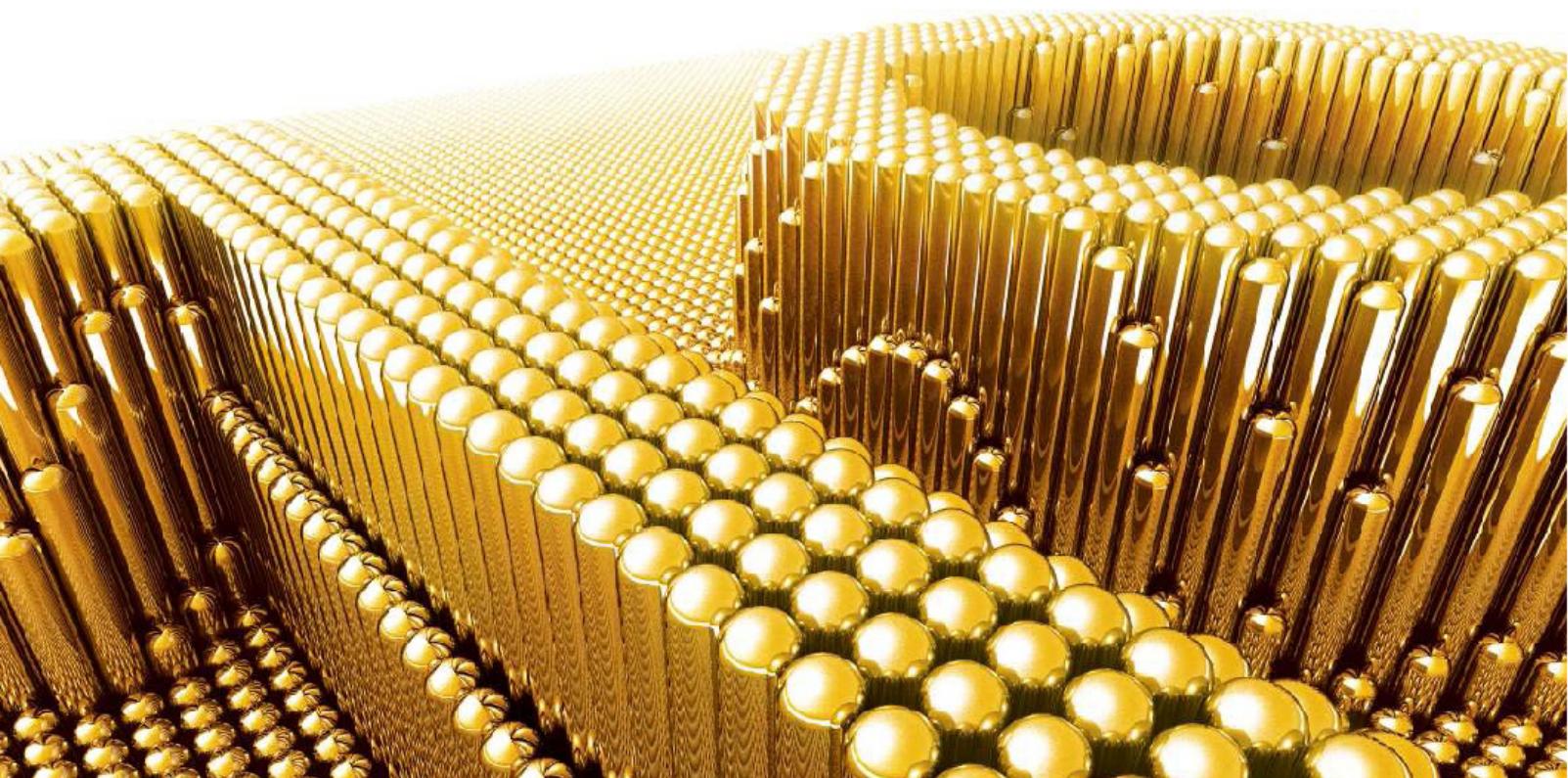




HALCON

the Power of Machine Vision

My hdev Example



EAN-13、EAN-13 Add-On 5、EAN-8、Code39、Code128、 MSI、RSS-14、RSS-14 Stacked、RSS-14 Stacked Omnidirectional、RSS Composite、RSS Expanded、RSS Expanded Stacked、RSS Limited 等类型的一维条形码读取的 代码实例

* Read bar codes of type EAN-13

```
*
```

```
create_bar_code_model ([][], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 120, 300, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_color ('green')
dev_set_draw ('margin')
dev_set_line_width (3)
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
set_bar_code_param (BarCodeHandle, 'element_size_max', 16)
for I := 1 to 15 by 1
    read_image (Image, 'barcode/ean13/ean13'+(I$'.2'))
    dev_resize_window_fit_image (Image, 0, 0, -1, -1)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
    area_center (SymbolRegions, Area, Row, Column)
    dev_display (Image)
    dev_display (SymbolRegions)
    disp_message (WindowHandle, DecodedDataStrings, 'image', Row-30, Column-90, 'black', 'true')
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
endfor
clear_bar_code_model (BarCodeHandle)
```

* Read bar code of type EAN-13 Add-On 5

```
*
```

```
create_bar_code_model ([][], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 120, 300, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_color ('green')
dev_set_draw ('margin')
dev_set_line_width (3)
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
set_bar_code_param (BarCodeHandle, 'element_size_max', 16)
read_image (Image, 'barcode/ean13addon5/ean13addon501')
dev_resize_window_fit_image (Image, 0, 0, -1, -1)
```

```
find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13 Add-On 5', DecodedDataStrings)
area_center (SymbolRegions, Area, Row, Column)
dev_display (Image)
dev_display (SymbolRegions)
disp_message (WindowHandle, DecodedDataStrings, 'image', Row-30, Column-90, 'black', 'true')
clear_bar_code_model (BarCodeHandle)
```

* Read bar codes of type EAN-8

```
*
```

```
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
read_image (Image, 'barcode/ean8/ean801')
dev_open_window_fit_image (Image, 0, 0, -1, -1, WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_color ('green')
dev_set_draw ('margin')
dev_set_line_width (3)
dev_display (Image)
find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-8', DecodedDataStrings)
disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
clear_bar_code_model (BarCodeHandle)
```

* Read bar codes of type Code 39

```
*
```

```
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 120, 300, 'black', WindowHandle)
dev_set_draw ('margin')
dev_set_line_width (3)
Colors := ['forest green','magenta']
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
for I := 1 to 5 by 1
    read_image (Image, 'barcode/code39/code39'+(I$'.2'))
    dev_resize_window_fit_image (Image, 0, 0, -1, -1)
    set_bar_code_param (BarCodeHandle, 'element_size_max', 6)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'Code 39', DecodedDataStrings)
    dev_display (Image)
    dev_display (SymbolRegions)
    get_bar_code_object (BarCodeObjects, BarCodeHandle, 'all', 'symbol_regions')
    get_bar_code_result (BarCodeHandle, 'all', 'decoded_strings', BarCodeResults)
    for J := 0 to |DecodedDataStrings|-1 by 1
        dev_set_color (Colors[J])
        select_obj (BarCodeObjects, ObjectSelected, J+1)
        dev_display (ObjectSelected)
    endfor
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, Colors, 'true')
    if (I < 5)
```

```
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
  endif
endfor
clear_bar_code_model (BarCodeHandle)
```

*** Read bar codes of type Code 128**

```
*
```

```
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
for I := 1 to 3 by 1
  read_image (Image, 'barcode/code128/code128'+(I$'.2'))
  dev_resize_window_fit_image (Image, 0, 0, -1, -1)
  find_bar_code (Image, SymbolRegions, BarCodeHandle, 'Code 128', DecodedDataStrings)
  get_bar_code_result (BarCodeHandle, 0, 'decoded_reference', Reference)
  String := "
  for J := 0 to strlen(DecodedDataStrings)-1 by 1
    if (ord(DecodedDataStrings{J}) < 32)
      Char := '\x'+ord(DecodedDataStrings{J})$'02x'
    else
      Char := DecodedDataStrings{J}
    endif
    String := String+Char
  endfor
  disp_message (WindowHandle, String, 'window', 12, 12, 'black', 'true')
  if (I < 3)
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
  endif
endfor
*
```

```
clear_bar_code_model (BarCodeHandle)
```

*** Read bar codes of type Codabar**

```
*
```

```
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 120, 300, 'black', WindowHandle)
dev_set_color ('green')
dev_set_draw ('margin')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
```

```

for I := 1 to 6 by 1
    read_image (Image, 'barcode/codabar/codabar'+(I$'.2'))
    get_image_size (Image, Width, Height)
    dev_set_window_extents (0, 0, Width-1, Height-1)
    dev_display (Image)
    set_bar_code_param (BarCodeHandle, 'check_char', 'present')
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'Codabar', DecodedDataStrings)
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'false')
    LastChar := strlen(DecodedDataStrings)-1
    disp_message      (WindowHandle,      DecodedDataStrings{0}+sum(gen_tuple_const(LastChar-1,'
'))+DecodedDataStrings{LastChar}, 'window', 12, 12, 'forest green', 'false')
    if (I < 6)
        stop ()
    endif
endfor
clear_bar_code_model (BarCodeHandle)

```

* Example program for decoding bar codes of type MSI

```

*
* Create Barcode reader with check character validation
create_bar_code_model ('check_char', 'present', BarCodeHandle)
*
* Initialization
dev_close_window ()
dev_open_window (0, 0, 320, 240, 'black', WindowHandle)
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*
* Read bar codes
for I := 1 to 3 by 1
    read_image (Image, 'barcode/msi/msi_' + I$'.2d')
    dev_resize_window_fit_image (Image, 0, 0, -1, -1)
    dev_display (Image)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'MSI', DecodedDataStrings)
    area_center (SymbolRegions, Area, Row, Column)
    dev_display (SymbolRegions)
    for IdxFound := 0 to |DecodedDataStrings|-1 by 1
        disp_message      (WindowHandle,      DecodedDataStrings[IdxFound],      'image',      Row[IdxFound]-20,
Column[IdxFound]-50, 'black', 'true')
    endfor
    if (I < 3)
        disp_continue_message (WindowHandle, 'black', 'true')
        stop ()
    endif
endfor
*
```

```
* Close Barcode reader  
clear_bar_code_model (BarCodeHandle)
```

* RSS-14 bar code symbology

```
*
```

* Create Barcode reader
create_bar_code_model ('element_size_min', 1.5, BarCodeHandle)
*

* Initialization
dev_close_window ()
dev_open_window (0, 0, 360, 256, 'black', WindowHandle)
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*

* Read bar codes
for I := 1 to 2 by 1
 read_image (Image, 'barcode/rss14/rss14_' + I\$'.2d')
 dev_display (Image)
 find_bar_code (Image, SymbolRegions, BarCodeHandle, 'RSS-14', DecodedDataStrings)
 get_region_runs (SymbolRegions, Row, ColumnBegin, ColumnEnd)
 dev_display (SymbolRegions)
 disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
 if (I < 2)
 disp_continue_message (WindowHandle, 'black', 'true')
 stop ()
endif
endfor
*

* Close Barcode reader
clear_bar_code_model (BarCodeHandle)

* RSS-14 Stacked bar code symbology

```
*
```

* Create Barcode reader
create_bar_code_model ([], [], BarCodeHandle)
*

* Initialization
dev_close_window ()
dev_open_window (0, 0, -1, -1, 'black', WindowHandle)
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*

* Read bar codes

```

for Ind := 1 to 2 by 1
  ImageName := 'barcode/rss14_stacked/rss14_stacked_+'+(Ind$'.2')
  read_image (Image, ImageName)
  get_image_size (Image, Width, Height)
  dev_set_window_extents (-1, -1, Width, Height)
  dev_display (Image)
  find_bar_code (Image, SymbolRegions, BarCodeHandle, 'RSS-14 Stacked', DecodedDataStrings)
  dev_display (SymbolRegions)
  disp_message (WindowHandle, DecodedDataStrings, 'image', 5, 5, 'black', 'true')
  if (Ind < 2)
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
  endif
endfor
*
* Close Barcode reader
clear_bar_code_model (BarCodeHandle)

```

*** RSS-14 Stacked Omnidirectional bar code symbology**

```

*
* Create Barcode reader
create_bar_code_model ([], [], BarCodeHandle)
*
* Initialization
dev_close_window ()
dev_open_window (0, 0, -1, -1, 'black', WindowHandle)
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*
* Read bar codes
for I := 1 to 2 by 1
  ImageName := 'barcode/rss14_stacked_omnidir/rss14_stacked_omnidir_+'+(I$'.2')
  read_image (Image, ImageName)
  get_image_size (Image, Width, Height)
  dev_set_window_extents (-1, -1, Width/2, Height/2)
  dev_display (Image)
  find_bar_code (Image, SymbolRegions, BarCodeHandle, 'RSS-14 Stacked Omnidir', DecodedDataStrings)
  dev_display (SymbolRegions)
  disp_message (WindowHandle, DecodedDataStrings, 'image', 5, 5, 'black', 'true')
  if (I < 2)
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
  endif
endfor
*
* Close Barcode reader

```

```

clear_bar_code_model (BarCodeHandle)

=====
* This program demonstrates the decoding of RSS bar codes with Composite component
*
dev_close_window ()
dev_open_window (0, 0, 512, 512, 'black', WindowHandle)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*
ScaleWindow := 1.0
*
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
*
create_bar_code_model ([], [], BarCodeHandle)
set_bar_code_param (BarCodeHandle, 'composite_code', 'CC-A/B')
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
*
TestParams := []
TestParams := [TestParams, 'rss14_stacked_composite_01', 'RSS-14 Stacked']
TestParams := [TestParams, 'rsslimited_composite_01',      'RSS Limited'    ]
TestParams := [TestParams, 'rsslimited_composite_02',      'RSS Limited'    ]
TestParams := [TestParams, 'rssexpanded_composite_01',     'RSS Expanded'   ]
*
for I := 0 to |TestParams|-2 by 2
    File := 'barcode/rss_composite/' + TestParams[I]
    CodeType := TestParams[I+1]
    *
    read_image (Image, File)
    * compensate for over exposed images: expand dark areas
    gray_erosion_shape (Image, Image, 1.5, 1.5, 'octagon')
    get_image_size (Image, Width, Height)
    dev_set_window_extents (0, 0, ScaleWindow*Width, ScaleWindow*Height)
    disp_message (WindowHandle, 'Barcode Type: ' + CodeType, 'image', 12, 12, 'black', 'true')
    *
    find_bar_code (Image, SymbolRegions, BarCodeHandle, CodeType, DecodedDataStrings)
    *
    get_bar_code_result (BarCodeHandle, 'all', 'decoded_strings', DecodedStrings)
    get_bar_code_result (BarCodeHandle, 'all', 'composite_strings', CompositeStrings)
    disp_message (WindowHandle, DecodedStrings + '\nComposite: ' + CompositeStrings, 'image', 36, 12,
    'black', 'true')
    if (I < |TestParams|-2)
        disp_continue_message (WindowHandle, 'black', 'true')
        stop ()
    endif
endfor
*
clear_bar_code_model (BarCodeHandle)

```

```
=====
* RSS Expanded bar code symbology
*
* Create Barcode reader
create_bar_code_model ('element_size_min', 1.5, BarCodeHandle)
*
* Initialization
NumImages := 2
dev_close_window ()
read_image (Image, 'barcode/rssexpanded/rssexpanded_01')
dev_open_window_fit_image (Image, 0, 0, -1, -1, WindowHandle)
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*
* Read bar codes
for I := 1 to NumImages by 1
    read_image (Image, 'barcode/rssexpanded/rssexpanded_' + I$.2d)
    dev_display (Image)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'RSS Expanded', DecodedDataStrings)
    dev_display (SymbolRegions)
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
    if (I < NumImages)
        disp_continue_message (WindowHandle, 'black', 'true')
        stop ()
    endif
endfor
*
* Close Barcode reader
clear_bar_code_model (BarCodeHandle)
```

* **RSS Expanded Stacked bar code symbology**

```
*
* Create Barcode reader
create_bar_code_model      (['element_size_max','element_size_min','max_diff_orient'],      [16,1.5,1.0],
BarCodeHandle)
*
* Initialization
dev_close_window ()
dev_open_window (0, 0, -1, -1, 'black', WindowHandle)
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
*
* Read bar codes
```

```

for Ind := 1 to 2 by 1
    ImageName := 'barcode/rssexpanded_stacked/rssexpanded_stacked_+' + (Ind$'.2')
    read_image (Image, ImageName)
    dev_resize_window_fit_image (Image, 0, 0, -1, -1)
    dev_display (Image)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'RSS Expanded Stacked', DecodedDataStrings)
    dev_display (SymbolRegions)
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
    if (Ind < 2)
        disp_continue_message (WindowHandle, 'black', 'true')
        stop ()
    endif
endfor
*
* Close Barcode reader
clear_bar_code_model (BarCodeHandle)
=====
```

*** RSS Limited bar code symbology**

```

*
* Create Barcode reader
create_bar_code_model ('element_size_min', 1.5, BarCodeHandle)
*
* Initialization
NumImages := 2
dev_close_window ()
dev_open_window (0, 0, 360, 256, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_draw ('margin')
dev_set_color ('green')
dev_set_line_width (3)
*
* Read bar codes
for I := 1 to NumImages by 1
    read_image (Image, 'barcode/rsslimited/rsslimited_' + I$'.2d')
    dev_display (Image)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'RSS Limited', DecodedDataStrings)
    dev_display (SymbolRegions)
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
    if (I < NumImages)
        disp_continue_message (WindowHandle, 'black', 'true')
        stop ()
    endif
endfor
*
* Close Barcode reader
clear_bar_code_model (BarCodeHandle)
```

Following bar code symbologies are supported:

2/5 Industrial	EAN-8	RSS-14	Code 39 - Full ASCII
2/5 Interleaved	EAN-8 Add-On 2	RSS-14 Truncated	Code 39 - Full ASCII
Codabar	EAN-8 Add-On 5	RSS-14 Stacked	Code 39 - Regular
Code 39	EAN-13	RSS-14 Stacked Omnidirectional	Code 93
Code 93	EAN-13 Add-On 2	RSS Limited	Code 128
Code 128	EAN-13 Add-On 5	RSS Expanded	Data Matrix
MSI	UPC-A	RSS Expanded Stacked	EAN/JAN-8
PharmaCode	UPC-A Add-On 2		EAN/JAN-13
	UPC-A Add-On 5		EAN/JAN/UPC 2-digit Supplemental
	UPC-E		EAN/JAN/UPC 5-digit Supplemental
	UPC-E Add-On 2		GS1 Composite
	UPC-E Add-On 5		GS1 DataBar (RSS)

GS1 数据矩阵
GS1-128
Interleaved 2-of-5
Japanese Post 4-State Customer Code
KIX 4-State Customer Code
MaxiCode
MicroPDF417
Micro QR Code
MSI Plessey
PDF417
QR Code
Royal Mail 4-State Customer Code
Telepen
UPC-A
UPC-E
USPS 智能邮件条形码 (4态客户代码)

Autodiscrimination describes the simultaneous decoding of multiple bar code types in one call of `find_bar_code`. For this purpose a tuple of bar code types is specified for the parameter `CodeType`. Using the generic value '`auto`' all known bar code types are decoded - except '`PharmaCode`' and '`MSI`' because these codes don't have enough features to be reliably separated from other bar code types. The tuple can also contain bar code types with the tilde prefix (~) which won't be decoded. For example

```
['auto', '~EAN-8', '~EAN-8 Add-On 2', '~EAN-8 Add-On 5']
```

describes all bar code types without '`PharmaCode`', '`MSI`' and all kinds of '`EAN-8`'. Please note that each additionally allowed bar code type increases the run-time of the operator. Using too many bar code types the reliability of the decoding could decrease because not all bar code types can be discriminated reliably. To improve autodiscrimination compatibility bar codes with a check character or check sum should be used.

The bar code reader tries to decode the bar code types in the following order:

RSS-14, RSS-14 Truncated, RSS-14 Stacked, RSS-14 Stacked Omnidirectional, RSS Limited, RSS Expanded, RSS Expanded Stacked, Code 128, EAN-13 Add-On 5, EAN-13 Add-On 2, EAN-13, UPC-A Add-On 5, UPC-A Add-On 2, UPC-A, EAN-8 Add-On 5, EAN-8 Add-On 2, EAN-8, UPC-E Add-On 5, UPC-E Add-On 2, UPC-E, Code 93, Code 39, Codabar, 2/5 Interleaved, 2/5 Industrial, MSI, PharmaCode.

Therefore you should exclude at least all definitely not occurring bar code types that are scanned before the first of the bar code types you expect to find or, better, just scan for the explicit list of bar code types you expect.

CodeType (input_control) string(-array) (string)

Type of the searched bar code.

Default value: 'EAN-13'

List of values: '2/5 Industrial', '2/5 Interleaved', 'Codabar', 'Code 39', 'Code 93', 'Code 128', 'EAN-13', 'EAN-13 Add-On 2', 'EAN-13 Add-On 5', 'EAN-8', 'EAN-8 Add-On 2', 'EAN-8 Add-On 5', 'UPC-A', 'UPC-A Add-On 2', 'UPC-A Add-On 5', 'UPC-E', 'UPC-E Add-On 2', 'UPC-E Add-On 5', 'MSI', 'PharmaCode', 'RSS-14', 'RSS-14 Truncated', 'RSS-14 Stacked', 'RSS-14 Stacked Omnidir', 'RSS Limited', 'RSS Expanded', 'RSS Expanded Stacked', 'auto'

```
=====
* Read bar codes of type 2/5 Industrial
*
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 120, 300, 'black', WindowHandle)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
dev_set_draw ('margin')
dev_set_line_width (3)
for I := 1 to 4 by 1
    read_image (Image, 'barcode/25industrial/25industrial0'+I)
    get_image_size (Image, Width, Height)
    dev_set_window_extents (0, 0, Width-1, Height-1)
    dev_display (Image)
    dev_set_color ('green')
    * Read bar code, the resulting string includes the check character
    set_bar_code_param (BarCodeHandle, 'check_char', 'absent')
    find_bar_code (Image, SymbolRegions, BarCodeHandle, '2/5 Industrial', DecodedDataStrings)
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'false')
    LastChar := strlen(DecodedDataStrings)-1
    disp_message (WindowHandle, sum(gen_tuple_const(LastChar, ' '))+DecodedDataStrings{LastChar},
    'window', 12, 12, 'forest green', 'false')
```

```
    stop ()
```

```
    * Read bar code using the check character to check the result, i.e.,
    * the check character does not belong to the returned string anymore.
    * If the check character is not correct, the bar code reading fails
```

```
    dev_set_color ('green')
```

```
    set_bar_code_param (BarCodeHandle, 'check_char', 'present')
    find_bar_code (Image, SymbolRegions, BarCodeHandle, '2/5 Industrial', DecodedDataStrings)
    disp_message (WindowHandle, DecodedDataStrings, 'window', 36, 12, 'black', 'false')
    dev_set_color ('magenta')
```

```
    if (I < 4)
```

```
        stop ()
```

```
    endif
```

```
endfor
```

```
clear_bar_code_model (BarCodeHandle)
```

* **Read bar codes of type 2/5 Interleaved**

```
*
```

```
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 120, 300, 'black', WindowHandle)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
dev_set_draw ('margin')
dev_set_line_width (3)
for I := 1 to 8 by 1
    read_image (Image, 'barcode/25interleaved/25interleaved'+(I$'.2'))
    get_image_size (Image, Width, Height)
    dev_set_window_extents (0, 0, Width-1, Height-1)
```

```
dev_display (Image)
dev_set_color ('green')
* Read bar code, the resulting string includes the check character
set_bar_code_param (BarCodeHandle, 'check_char', 'absent')
find_bar_code (Image, SymbolRegions, BarCodeHandle, '2/5 Interleaved', DecodedDataStrings)
disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'false')
LastChar := strlen(DecodedDataStrings)-1
disp_message (WindowHandle, sum(gen_tuple_const(LastChar, ' '))+DecodedDataStrings{LastChar},
>window', 12, 12, 'forest green', 'false')
stop ()
* Read bar code using the check character to check the result, i.e.,
* the check character does not belong to the returned string anymore.
* If the check character is not correct, the bar code reading fails
dev_set_color ('green')
set_bar_code_param (BarCodeHandle, 'check_char', 'present')
find_bar_code (Image, SymbolRegions, BarCodeHandle, '2/5 Interleaved', DecodedDataStrings)
disp_message (WindowHandle, DecodedDataStrings, 'window', 36, 12, 'black', 'false')
if (I<8)
    stop ()
endif
endfor
clear_bar_code_model (BarCodeHandle)
=====
```

Stage I's hdev 的代码实例

* Autodiscrimination A.hdev

*

* Code generated by Image Acquisition 03

* 获取条形码，并计算及显示解码时间，并从规定读取的条码类型范围中读取被解码的条码类型，

* 也可设置成不规定条码类型即自动识别，但会增加解码时间，甚至出现误读的情况。

* 可读取多个不同类型的条码，并且多个条码用不同颜色的区域框区分，且读取出来的信息也以

* 相应的颜色做区分

dev_close_window ()

dev_open_window (0, 0, 600, 600, 'black', WindowHandle)

*先关闭活动图形窗口，再打开这个窗口，标识符为 WindowHandle；

*相对于界面左上角第 0 行、第 0 列，大小为 300×300 像素，颜色为黑色。

open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0, -1, AcqHandle)

* 打开帧接收器（图像采集设备，如摄像头，工业相机等），参数（Parameter）详见这个算子

* 注意，采集器名称不同要更改，或者用助手获取也可以。

grab_image_start (AcqHandle, -1)

while (true)

grab_image_async (Image, AcqHandle, -1)

*开始条形码识别

create_bar_code_model ([], [], BarCodeHandle)

*必备的创建条码解码的开头，下面有一段结束代码

dev_update_var ('off')

dev_update_pc ('off')

dev_update_window ('off')

* 刷新窗体

set_display_font (WindowHandle, 14, 'mono', 'true', 'false')

dev_set_draw ('margin')

dev_set_line_width (3)

Colors := ['forest green','magenta','blue','red','yellow']

* 设置区域框的属性，如无填充色、边框线大小为 3，颜色。

*CodeTypes := ['auto']

CodeTypes := ['EAN-13','Code 39','EAN-8','Code 128','Code 93']

* 设置条码的类型，如 EAN-13 等 5 种一维码类型，或者 auto 自动获取（但解码时间较长且有误读的可能）

get_image_size (Image, Width, Height)

dev_display (Image)

*获取图像大小，并显示图像

count_seconds (Start)

find_bar_code (Image, SymbolRegions, BarCodeHandle, CodeTypes, DecodedDataStrings)

count_seconds (Stop)

Duration := (Stop - Start) * 1000

*开始计时、寻找并解码条形码（参数解释详见这个算子），结束计时，并计算解码的时间

dev_display (SymbolRegions)

get_bar_code_object (BarCodeObjects, BarCodeHandle, 'all', 'symbol_regions')

get_bar_code_result (BarCodeHandle, 'all', 'decoded_types', DecodedDataTypes)

*area_center (SymbolRegions, Area, Rows, Columns)

```

for J := 0 to |DecodedDataStrings|-1 by 1
    dev_set_color (Colors[J])
    select_obj (BarCodeObjects, ObjectSelected, J+1)
    dev_display (ObjectSelected)
endfor
*循环获取，并且根据获取的条码数量，设置区域框的颜色
if (|DecodedDataStrings| >= 1)
    *disp_message (WindowHandle, ['Found bar code(s) in ' + Duration$'3.0f' + 'ms:', '\n Type: ' +
DecodedDataTypes + '\n Data: ' + DecodedDataStrings], 'window', 5*12, 12, 'black', 'true')
        disp_message (WindowHandle, '读取条形码,用时 {' + Duration$'3.0f' + 'ms}: ' + '[' +
DecodedDataTypes + ']' + '=' + ' " ' + DecodedDataStrings + '" ', 'window', 12, 12, Colors, 'true')
        disp_continue_message (WindowHandle, 'black', 'true')
        wait_seconds(3)
endif
*判断读取到数据的数量，当大于 1 (数字可改为 2、3……) 个条码被解码时，显示信息在窗体
的第 12 行，第 12 列的位置，并按顺序用不同的颜色标注
clear_bar_code_model (BarCodeHandle)
*关闭条形码解码
endwhile
close_framegrabber (AcqHandle)
*关闭帧接收器 (摄像头)

```

* Autodiscrimination B.hdev

```

*
* Code generated by Image Acquisition 04
* 寻找正确的、所需的条码类型的条形码，当寻找到时，hold 住图像
dev_close_window ()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
*先关闭活动图形窗口，再打开这个窗口，标识符为 WindowHandle;
*相对于界面左上角第 0 行、第 0 列，大小为 300×300 像素，颜色为黑色。
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
*注意摄像头的名称
grab_image_start (AcqHandle, -1)
while (true)
    grab_image_async (Image, AcqHandle, -1)
    *开始条形码识别
    create_bar_code_model ([], [], BarCodeHandle)
        dev_update_var ('off')
        dev_update_pc ('off')
        dev_update_window ('off')
        set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
        dev_set_draw ('margin')
        dev_set_line_width (3)
        dev_set_color ('forest green')
    *Colors := ['forest green','magenta','blue','red','yellow']
    CodeTypes := ['EAN-13','Code 39','EAN-8','Code 128','Code 93']
    *CodeTypes := ['auto']

```

```

get_image_size (Image, Width, Height)
dev_display (Image)
count_seconds (Start)
find_bar_code (Image, SymbolRegions, BarCodeHandle, CodeTypes, DecodedDataStrings)
count_seconds (Stop)
Duration := (Stop - Start) * 1000
dev_display (SymbolRegions)
get_bar_code_result (BarCodeHandle, 'all', 'decoded_types', DecodedDataTypes)
area_center (SymbolRegions, Area, Rows, Columns)
for IdxResult := 0 to |DecodedDataStrings|-1 by 1
    if (regexp_test (DecodedDataTypes[IdxResult], 'EAN-13') # 0)
        Color := 'forest green'
    else
        Color := 'red'
    endif
    dev_set_color (Color)
    select_obj (SymbolRegions, SelectedRegion, IdxResult+1)
    dev_display (SelectedRegion)
    disp_message (WindowHandle, DecodedDataTypes[IdxResult]+': '+DecodedDataStrings[IdxResult],
'image', Rows[IdxResult], Columns[IdxResult]-160, Color, 'true')
endfor
*if (|DecodedDataStrings| >= 2)
if (DecodedDataTypes = 'EAN-13')
    *disp_message (WindowHandle, ['Found bar code(s) in ' + Duration$'3.0f' + 'ms:', '\n Type: ' +
DecodedDataTypes + '\n Data: ' + DecodedDataStrings], 'window', 5*12, 12, 'black', 'true')
    *disp_message (WindowHandle, '读取条形码,用时{' + Duration$'3.0f' + 'ms}:[' + '[' +
DecodedDataTypes + ']' + '=' + '“' + DecodedDataStrings + '” ', 'window', 12, 12, Colors, 'true')
    disp_continue_message (WindowHandle, 'black', 'true')
    stop()
endif
*endif
clear_bar_code_model (BarCodeHandle)
endwhile
close_framegrabber (AcqHandle)

```

* Autodiscrimination C.hdev

```

*
* Code generated by Image Acquisition 05
* 读取每个不同的条码时，显示不同颜色的区域框，获取的条码信息也为相应的区域边框颜色
dev_close_window ()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
*先关闭活动图形窗口，再打开这个窗口，标识符为 WindowHandle;
*相对于界面左上角第 0 行、第 0 列，大小为 300×300 像素，颜色为黑色。
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
*注意摄像头的名称
grab_image_start (AcqHandle, -1)
while (true)

```

```

grab_image_async (Image, AcqHandle, -1)
*开始条形码识别
create_bar_code_model ([], [], BarCodeHandle)
    dev_update_var ('off')
    dev_update_pc ('off')
    dev_update_window ('off')
    set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
    dev_set_draw ('margin')
    dev_set_line_width (3)
    *dev_set_color ('forest green')
    Colors := ['forest green','magenta','blue','red','yellow']
    CodeTypes := ['EAN-13','Code 39','EAN-8','Code 128','Code 93']
    *CodeTypes := ['auto']
    get_image_size (Image, Width, Height)
    dev_display (Image)
    count_seconds (Start)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, CodeTypes, DecodedDataStrings)
    count_seconds (Stop)
    Duration := (Stop - Start) * 1000
    dev_display (SymbolRegions)
    get_bar_code_result (BarCodeHandle, 'all', 'decoded_types', DecodedDataTypes)
    area_center (SymbolRegions, Area, Rows, Columns)
    for IdxResult := 0 to |DecodedDataStrings|-1 by 1

        dev_set_color (Colors[IdxResult])
        select_obj (SymbolRegions, SelectedRegion, IdxResult+1)
        dev_display (SelectedRegion)
        disp_message (WindowHandle, DecodedDataTypes[IdxResult]+': '+DecodedDataStrings[IdxResult],
        'image', Rows[IdxResult], Columns[IdxResult]-160, Colors[IdxResult], 'true')
    endfor
    if (|DecodedDataStrings| >= 2)
        *if (DecodedDataTypes = 'EAN-13')
            *disp_message (WindowHandle, ['Found bar code(s) in ' + Duration$'3.0f' + 'ms:', '\n Type: ' +
            DecodedDataTypes + '\n Data: ' + DecodedDataStrings], 'window', 5*12, 12, 'black', 'true')
            *disp_message (WindowHandle, '读取条形码,用时 {' + Duration$'3.0f' + 'ms}: ' + '[' +
            DecodedDataTypes + ']' + ' ' + DecodedDataStrings + ' ', 'window', 12, 12, Colors, 'true')
            disp_continue_message (WindowHandle, 'black', 'true')
            stop()
        endif
        *endif
    clear_bar_code_model (BarCodeHandle)
endwhile
close_framegrabber (AcqHandle)

```

* Autodiscrimination D.hdev

*

* Code generated by Image Acquisition 06

* 读取条码的方向值，但是值是为什么是这个数值，还不清楚。

```

dev_close_window()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
*先关闭活动图形窗口，再打开这个窗口，标识符为WindowHandle;
*相对于界面左上角第 0 行、第 0 列，大小为 300×300 像素，颜色为黑色。
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
*注意摄像头的名称
grab_image_start (AcqHandle, -1)
while (true)
    grab_image_async (Image, AcqHandle, -1)
    *开始条形码识别
    create_bar_code_model ([], [], BarCodeHandle)
        dev_display (Image)
        dev_set_draw ('margin')
        dev_set_line_width (3)
        dev_set_color ('green')
        set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
        get_bar_code_param (BarCodeHandle, 'orientation', InputOrient)
        get_bar_code_param (BarCodeHandle, 'orientation_tol', InputOrientTol)
        Mode := 'Default orientation'
        set_bar_code_param (BarCodeHandle, ['orientation','orientation_tol'], [InputOrient,InputOrientTol])
        count_seconds (Seconds1)
        find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
        count_seconds (Seconds2)
        Time := (Seconds2-Seconds1)*1000
        dev_display (Image)
        dev_display (SymbolRegions)
        get_bar_code_result (BarCodeHandle, 'all', 'orientation', Orientation)
        area_center (SymbolRegions, Area, Row, Col)
        *gen_arrow_contour_xld      (Arrow,      Row+sin(rad(Orientation))*70,      Col-cos(rad(Orientation))*70,
Row-sin(rad(Orientation))*70, Col+cos(rad(Orientation))*70, 25, 25)
        *dev_display (Arrow)
        * 暂时获取方向的箭头 Arrow，只能获取一个静态的，但动态时，就出错？
        for I := 0 to |Row|-1 by 1
            disp_message (WindowHandle, 'Orientation = ' + Orientation[I] $ '.1f'+chr(0xb0), 'image',
Row[I]-30, Col[I]-120, 'black', 'true')
        endfor
        Message := Mode + ': '+InputOrient+chr(0xb0)+'+/- '+InputOrientTol+chr(0xb0)
        Message[1] := |Row|+' code(s) found in ' + Time$.1f' + ' ms'
        disp_message (WindowHandle, Message, 'window', 12, 12, 'black', 'true')
        if (|Orientation| >= 1)
            disp_continue_message (WindowHandle, 'black', 'true')
            stop ()
        endif
        clear_bar_code_model (BarCodeHandle)
    endwhile
    close_framegrabber (AcqHandle)

```

```
=====
* Camera 2D Code Read.hdev
*
* 2D Code generated by Image Acquisition 01
* QR Code
dev_close_window()
dev_open_window(0, 0, 400, 400, 'black', WindowHandle)
*先关闭活动图形窗口，再打开这个窗口，标识符为WindowHandle;
*相对于界面左上角第0行、第0列，大小为400×400像素，颜色为黑色。
open_framegrabber('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
* 打开帧接收器（图像采集设备，如摄像头，工业相机等），参数（Parameter）详见这个算子
* 注意摄像头的名称，可以用工具栏中的“助手”——打开新的Image Acquisition 获取摄像头及插入代码
grab_image_start(AcqHandle, -1)
while (true)
    grab_image_async(Image, AcqHandle, -1)
    create_data_code_2d_model('QR Code', [], [], DataCodeHandle)
    * 二维码的创建开头的算子，clear为结束清除的算子，见下。
    set_display_font(WindowHandle, 16, 'mono', 'true', 'false')
    dev_set_color('forest green')
    dev_set_draw('margin')
    dev_set_line_width(3)
    set_data_code_2d_param(DataCodeHandle, 'default_parameters', 'enhanced_recognition')
    * 设置选定参数的二维数据模型，参数详见这个算子
    find_data_code_2d(Image, SymbolXLDs, DataCodeHandle, [], [], ResultHandles, DecodedDataStrings)
    * 检测和读取二维代码符号，也支持读取二维数据模型的序列，参数详见这个算子
    for i := 0 to |ResultHandles| - 1 by 1
        select_obj(SymbolXLDs, SymbolXLD, i+1)
        get_contour_xld(SymbolXLD, Row, Col)
        get_string_extents(WindowHandle, DecodedDataStrings[i], Ascent, Descent, TxtWidth, TxtHeight)
        disp_message (WindowHandle, DecodedDataStrings[i], 'image', max(Row-50),
max([min(Col+30)-TxtWidth/2, 1]), 'black', 'true')
    endfor
    *这段for循环语句的目的是让解码到的字符串（二维码的内容）显示到二维码深绿色（forest green
上面定义）的解码区域框的行列位置。
    *disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
    *如果不设置显示到区域框中间的位置，而是显示到窗体的上方或其他位置，那么不需要上面
那段for语句，只需这段信息显示的语句即可显示到窗体相应位置。
    if (|DecodedDataStrings|>0)
        disp_continue_message (WindowHandle, 'black', 'true')
        stop()
    endif
    *if语句，当解码一个（大于0，可设置多个）二维码就暂停摄像头获取图像，直至按F5。
    clear_data_code_2d_model(DataCodeHandle)
endwhile
close_framegrabber(AcqHandle)
```

```
=====
* Camera Bar Code Read.hdev
*
* Code generated by Image Acquisition 01
dev_close_window ()
dev_open_window (0, 0, 300, 300, 'black', WindowHandle)
*先关闭活动图形窗口，再打开这个窗口，标识符为WindowHandle;
*相对于界面左上角第0行、第0列，大小为300×300像素，颜色为黑色。
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
*注意摄像头的名称
grab_image_start (AcqHandle, -1)
while (true)
    grab_image_async (Image, AcqHandle, -1)
    create_bar_code_model ([], [], BarCodeHandle)
        set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
        dev_set_color ('orange red')
        *设置读取条形码区域框的边框颜色，如橘红色。
        dev_set_draw ('margin')
        *设置读取条形码区域框的填充色，fill全部填充上句设置的橘红色，margin仅留边框颜色不填充颜色。
        dev_set_line_width (3)
        *设置读取条形码区域框的边框线条粗细度。
        set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
        set_bar_code_param (BarCodeHandle, 'element_size_max', 16)
        dev_resize_window_fit_image (Image, 0, 0, -1, -1)
        find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
        area_center (SymbolRegions, Area, Row, Column)
        dev_display (Image)
        dev_display (SymbolRegions)
        if (|DecodedDataStrings|>0)
            disp_message (WindowHandle, DecodedDataStrings, 'image', Row-30, Column-90, 'black', 'true')
            disp_continue_message (WindowHandle, 'black', 'true')
            stop()
        endif
    clear_bar_code_model (BarCodeHandle)
endwhile
close_framegrabber (AcqHandle)
```

```
=====
* Camera Read and Stop with the 1 to 3 barcode.hdev
*
* Code generated by Image Acquisition 02
dev_close_window ()
dev_open_window (0, 0, 300, 300, 'black', WindowHandle)
*先关闭活动图形窗口，再打开这个窗口，标识符为WindowHandle;
*相对于界面左上角第0行、第0列，大小为300×300像素，颜色为黑色。
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
```

```

grab_image_start (AcqHandle, -1)
while (true)
    grab_image_async (Image, AcqHandle, -1)
    *开始条形码识别
        create_bar_code_model ([], [], BarCodeHandle)
        dev_set_draw ('margin')
        dev_set_line_width (4)
        Colors := ['forest green','magenta','blue','red','yellow']
        set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
        dev_resize_window_fit_image (Image, 0, 0, -1, -1)
        set_bar_code_param (BarCodeHandle, 'element_size_max', 6)
        find_bar_code (Image, SymbolRegions, BarCodeHandle, ['EAN-13','EAN-8','Code 39','Code 128'],
DecodedDataStrings)
        dev_display (Image)
        dev_display (SymbolRegions)
        get_bar_code_object (BarCodeObjects, BarCodeHandle, 'all', 'symbol_regions')
        get_bar_code_result (BarCodeHandle, 'all', 'decoded_strings', BarCodeResults)
        for J := 0 to |DecodedDataStrings|-1 by 1
            dev_set_color (Colors[J])
            select_obj (BarCodeObjects, ObjectSelected, J+1)
            dev_display (ObjectSelected)
        endfor
        if (|DecodedDataStrings| >= 2)
            disp_message (WindowHandle, 'Found bar code: ' + DecodedDataStrings, 'window', 12, 12,
Colors, 'true')
            disp_continue_message (WindowHandle, 'black', 'true')
            stop()
        endif
        clear_bar_code_model (BarCodeHandle)
    endwhile
    close_framegrabber (AcqHandle)

```

* color set.hdev

- *
- * 在 set_color 的算子中，列出 Halcon 支持的颜色代码，共 21 种。
- * Suggested values: 'black', 'white', 'red', 'green',
- * 'blue', 'cyan', 'magenta', 'yellow', 'dim gray', 'gray',
- * 'light gray', 'medium slate blue', 'coral', 'slate blue',
- * 'spring green', 'orange red', 'orange', 'dark olive green',
- * 'pink', 'forest green', 'cadet blue'
- * 可以用以下的代码来测试颜色

```

ColorSet:={}
ColorSet[1]:='gray'
ColorSet[2]:='magenta'
ColorSet[3]:='dim gray'
ColorSet[4]:='coral'
ColorSet[5]:='slate blue'
ColorSet[6]:='spring green'

```

```

ColorSet[7]:='orange red'
ColorSet[8]:='cadet blue'
ColorSet[9]:='light gray'
ColorSet[10]:='medium slate blue'
ColorSet[11]:='red'
ColorSet[12]:='white'
ColorSet[13]:='green'
ColorSet[14]:='blue'
ColorSet[15]:='yellow'
ColorSet[16]:='pink'
ColorSet[17]:='orange'
ColorSet[18]:='cyan'
ColorSet[19]:='black'

```

for i:=1 to 19 by 1

```

    dev_update_window('off')
    dev_close_window()
    dev_open_window(0,0,300,300,ColorSet[i],WindowHandle)
    get_system ('operating_system', OS)
    set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
    disp_message (WindowHandle, ['The Color is:',ColorSet[i] ], 'window', -1, -1, [ColorSet[i],ColorSet[i]], 'true')
    wait_seconds(1)

```

endfor

* EAN-13 Bar code read.hdev

```

*
* Read bar codes of type EAN-13
*
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 300, 300, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_color ('green')
dev_set_draw ('margin')
dev_set_line_width (3)
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
set_bar_code_param (BarCodeHandle, 'element_size_max', 16)
for I := 1 to 15 by 1
    read_image (Image,'L:/Halcon test/barcode/image/' + I + '.png')
    * 注意修改图片路径
    dev_resize_window_fit_image (Image, 0, 0, -1, -1)
    find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
    area_center (SymbolRegions, Area, Row, Column)
    dev_display (Image)

```

```

dev_display (SymbolRegions)
disp_message (WindowHandle, DecodedDataStrings, 'image', Row-30, Column-90, 'black', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
endfor
clear_bar_code_model (BarCodeHandle)

=====
* image acquisition.hdev
*
*=====分割线 1=====
*图像获取的基本代码 (1)
*读取路径下的图片,image1
read_image(image1,'L:/Halcon test/barcode/image/1.png')

*=====分割线 2=====
*图像获取的基本代码 (2)
*读取路径下的多张图片,image2
imagepath :=[]
*建立一个图像读取的数组，命名为 imagepath。
imagepath[0]:='L:/Halcon test/barcode/image/2.png'
imagepath[1]:='L:/Halcon test/barcode/image/3.png'
imagepath[2]:='L:/Halcon test/barcode/image/4.png'
imagepath[3]:='L:/Halcon test/barcode/image/5.png'
*读取 4 张图片的路径
for i:=0 to 3 by 1
    read_image(image2,imagepath[i])
endfor

*=====分割线 3=====
*图像获取的基本代码 (3)
*读取文件夹下的某些图片,image3
for i:=1 to 5 by 1
    read_image(image3,'L:/Halcon test/barcode/image/' + i + '.png')
endfor

*=====分割线 4=====
*图像获取的基本代码 (4)
*读取文件夹下的所有图片和文件夹， image4
* Code generated by Image Acquisition 01
list_files ('L:/Halcon test/barcode/image', ['files','directories','recursive','max_depth 2','follow_links'], ImageFiles)
tuple_regexp_select (ImageFiles, ['\.(tif|tiff|gif|bmp|jpg|jpeg|jp2|png|pcx|pgm|ppm|pbm|xwd|ima)$','ignore_case'], ImageFiles)
* 'ignore_case': 忽略大小写
* '\.' --> '!'
for Index := 0 to |ImageFiles| - 1 by 1
    read_image (Image4, ImageFiles[Index])
    * Do something
endfor

```

```
*=====分割线 5=====

*图像获取的基本代码 (5)
*连接摄像机读取图像, image5
* Code generated by Image Acquisition 02
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', '0', -1, -1, AcqHandle)
grab_image_start (AcqHandle, -1)
while (true)
    grab_image_async (Image5, AcqHandle, -1)
    * Do something
endwhile
close_framegrabber (AcqHandle)
```

* one of EAN-13 Bar code read.hdev

```
*
```

```
* Read bar codes of type EAN-13
*
create_bar_code_model ([], [], BarCodeHandle)
dev_close_window ()
dev_open_window (0, 0, 300, 600, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_color ('green')
dev_set_draw ('margin')
dev_set_line_width (3)
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
set_bar_code_param (BarCodeHandle, 'element_size_max', 16)
read_image (Image, 'F:/Halcon test/test/test201211290058.jpg')
* 注意修改图片路径
dev_resize_window_fit_image (Image, 0, 0, 300, 600)
find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
area_center (SymbolRegions, Area, Row, Column)
dev_display (Image)
dev_display (SymbolRegions)
disp_message (WindowHandle, DecodedDataStrings, 'image', Row-30, Column-90, 'black', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
clear_bar_code_model (BarCodeHandle)
```

* visualization of bar code scanlines with Camera.hdev

```
*
```

```
* This program demonstrates visualization of bar code scanlines.
* 这个程序演示了可视化的条码扫描线。
* This visualization can be used to inspect the quality of images, where
* occlusions or print defects might prevent the bar code reader from
* successfully decoding the underlying bar code.
* 这种可视化可用于检查图像的质量, 其中封闭遮断或打印缺陷可能会阻止条形码阅读器成功解码潜在的条码。
```

- * With 'scanlines_all' all scanlines that the bar code reader would eventually use to decode a candidate bar code (here, colored in red).
- * 'scanlines_all' 用于显示标识最终的解码潜在条码的扫描线（红色部分）
- * With 'scanlines_valid' are visualized all scanlines that can be decoded as well (here, colored in green)
- * 'scanlines_valid' 用于显示标识有效解码的扫描线
- * 'candidate_regions' 用于获取条码潜在区域

```

dev_close_window()
dev_open_window(0, 0, 500, 500, 'black', WindowHandle)
set_display_font(WindowHandle, 16, 'mono', 'true', 'false')
dev_set_draw('margin')
dev_update_window('off')
dev_set_line_width(2)
open_framegrabber('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0, -1, AcqHandle)
grab_image_start(AcqHandle, -1)
while(true)
create_bar_code_model([], [], BarCodeHandle)
set_bar_code_param(BarCodeHandle, 'element_size_min', 1.5)
*这个参数是指条码的条（黑线）和空（空白间隔）的最小宽度，对于尺寸很大的条码，这个最小值 element_size_min
*要相应增加，比如一个较大的条码图片，用 1.5 可能就读不到，因为条和空的宽度很大，用 2 或 2.5 以上就可以解码，
*但数值越大，解码的时间就越长。还有一个 element_size_max 的参数，和这个相反，属于限制最大宽度，宽度不能太大，
*不然相邻的条和空就会被识别成一个了。一般这两个可以一起用
    grab_image_async(Image, AcqHandle, -1)
    set_bar_code_param(BarCodeHandle, 'persistence', 1)
    * persistence 这个参数没有查到，暂未知
        get_image_size(Image, Width, Height)
        dev_set_window_extents(0, 0, Width, Height)
        dev_display(Image)
        *
        find_bar_code(Image, SymbolRegions, BarCodeHandle, 'Code 128', DecodedDataStrings)
        * 寻找条码，并获取解码值。
        get_bar_code_object(Candidates, BarCodeHandle, 'all', 'candidate_regions')
        dev_set_color('magenta')
        dev_display(Candidates)
        * 获取条码的扫描线候选区域。
        get_bar_code_object(AllScanlines, BarCodeHandle, 'all', 'scanlines_all')
        dev_set_color('red')
        dev_display(AllScanlines)
        * 获取所有的扫描线（扫描线应该是指在一定的高度内读取到的条和空的边缘形成的线），这个读取不论是否能解码，均读取显示。
        get_bar_code_object(ValidScanlines, BarCodeHandle, 'all', 'scanlines_valid')
        dev_set_color('green')
        dev_display(ValidScanlines)
        * 获取有效的条码扫描线
        if (|DecodedDataStrings|>0)

```

```
    disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
endif
* if 语句，当读到了数值就停下来。
```

```
clear_bar_code_model (BarCodeHandle)
endwhile
close_framegrabber (AcqHandle)
*关闭帧接收器（摄像头）
```

* visualization of bar code scanlines.hdev

```
*
* This program demonstrates visualization of bar code scanlines.
* 这个程序演示了可视化的条码扫描线。
* This visualization can be used to inspect the quality of images, where
* occlusions or print defects might prevent the bar code reader from
* successfully decoding the underlying bar code.
* 这种可视化可用于检查图像的质量，其中封闭遮断或打印缺陷可能会阻止条形码阅读器成功解码潜在的条码。
* With 'scanlines_all' all scanlines that the bar code reader would
* eventually use to decode a candidate bar code (here, colored in red).
* 'scanlines_all'用于显示标识最终的解码潜在条码的扫面线（红色部分）
* With 'scanlines_valid' are visualized all scanlines that can be decoded
* as well (here, colored in green)
* 'scanlines_valid'用于显示标识有效解码的扫描线
* 'candidate_regions'用于获取条码潜在区域
```

```
dev_close_window ()
dev_open_window (0, 0, 500, 500, 'black', WindowHandle)
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
dev_set_draw ('margin')
dev_update_window ('off')
dev_set_line_width (2)
```

```
create_bar_code_model ([], [], BarCodeHandle)
```

```
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
```

*这个参数是指条码的条（黑线）和空（空白间隔）的最小宽度，对于尺寸很大的条码，这个最小值 element_size_min

*要相应增加，比如一个较大的条码图片，用 1.5 可能就读不到，因为条和空的宽度很大，用 2 或 2.5 以上就可以解码，

*但数值越大，解码的时间就越长。还有一个 element_size_max 的参数，和这个相反，属于限制最大宽度，宽度不能太大，

*不然相邻的条和空就会被识别成一个了。一般这两个可以一起用

```
set_bar_code_param (BarCodeHandle, 'persistence', 1)
```

* persistence 这个参数没有查到，暂未知

```
read_image (Image, L:/Halcon test/barcode/test 020.jpg)
get_image_size (Image, Width, Height)
dev_set_window_extents (0, 0, Width, Height)
```

```

dev_display (Image)
*
find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
* 寻找条码，并获取解码值。
get_bar_code_object (Candidates, BarCodeHandle, 'all', 'candidate_regions')
dev_set_color ('magenta')
dev_display (Candidates)
* 获取条码的扫描线候选区域。
get_bar_code_object (AllScanlines, BarCodeHandle, 'all', 'scanlines_all')
dev_set_color ('red')
dev_display (AllScanlines)
* 获取所有的扫描线（扫描线应该是指在一定的高度内读取到的条和空的边缘形成的线），这个读取不论是否能解码，均读取显示。
get_bar_code_object (ValidScanlines, BarCodeHandle, 'all', 'scanlines_valid')
dev_set_color ('green')
dev_display (ValidScanlines)
* 获取有效的条码扫描线
if (|DecodedDataStrings|>0)
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
endif
* if 语句，当读到了数值就停下来。

```

```
clear_bar_code_model (BarCodeHandle)
```

* 物体跟踪程序.hdev

```

* Code generated by Image Acquisition 01
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0,
-1, AcqHandle)
grab_image_start (AcqHandle, -1)
while (true)
    grab_image_async (Image, AcqHandle, -1)
    get_image_size (Image, Width, Height)
    *获取图像尺寸
    dev_open_window (0, 0, Width, Height, 'black', WindowID)
    *打开窗口
    set_display_font (WindowID, 14, 'mono', 'true', 'false')
    *设置系统字体
    dev_set_draw ('margin')
    *定义区域填充模式
    dev_set_colored (12)
    *显示 region 用到的颜色数目
    dev_set_line_width (3)
    *设置线条宽度
    dev_display (Image)
    *显示图像
    zoom_image_factor (Image, ImageZoomed, 0.5, 0.5, 'constant')

```

```

*图像缩放规定因子倍
create_bg_esti (ImageZoomed, 0.7, 0.7, 'fixed', 0.001, 0.03, 'on', 8.0, 10, 3.25, 15, BgEstiHandle)
*生成和初试化一组用于背景估计的数据集
while (true)
    *循环语句
    grab_image_async (ActualImage, AcqHandle, -1)
    *抓拍图像
    zoom_image_factor (ActualImage, ImageZoomed, 0.5, 0.5, 'constant')
    *图像缩放规定因子倍
    run_bg_esti (ImageZoomed, ForegroundRegion, BgEstiHandle)
    *估计背景和提取前景
    zoom_region (ForegroundRegion, RegionZoom, 2, 2)
    *缩放一个区域
    intersection (RegionZoom, RegionZoom, RegionIntersection)
    *计算两个区域的交集
    connection (RegionIntersection, ConnectedRegions)
    *计算一个区域相连接的部分
    select_shape (ConnectedRegions, SelectedRegions, 'area', 'and', 20, 99999)
    *选择带有某些特征的区域
    shape_trans (SelectedRegions, RegionTrans, 'convex')
    *变换形状
    union1 (RegionTrans, RegionUnion)
    *返回所有输入区域的并集
    connection (RegionUnion, FinalConnected)
    *计算一个区域相连接的部分
    select_gray (FinalConnected, ActualImage, SelectedRegionsGray, 'deviation', 'and', 25, 1000)
    *根据灰度值的特点来选择区域
    union1 (SelectedRegionsGray, RegionUnionFinal)
    *返回所有输入区域的并集
    dilation_rectangle1 (RegionUnionFinal, RegionDilation, 5, 5)
    *扩大一个矩形结构基础的一个区域
    complement (RegionDilation, BackGround)
    *返回一个区域的补码
    clip_region (BackGround, RegionClipped, 0, 0, Height-1, Width-1)
    *将一个区域修改为矩形
    zoom_region (RegionClipped, BackGroundZoom, 0.5, 0.5)
    *缩放一个区域
    update_bg_esti (ImageZoomed, BackGroundZoom, BgEstiHandle)
    *更新被估计的背景图像数据集
    shape_trans (SelectedRegionsGray, Rects, 'rectangle1')
    *改变一个区域的形状
    dev_display (ActualImage)
    *在图形窗口显示输入的图形对象
    dev_display (Rects)
    *在图形窗口显示输入的图形对象
endwhile
endwhile
close_framegrabber (AcqHandle)

```

一个用摄像头获取条形码并显示类型及解码的实例及每个步骤注解

分享一个用摄像头获取条形码并显示类型及解码的实例及每个步骤注解

* Code generated by Image Acquisition 03

* 获取条形码，并计算及显示解码时间，并从规定读取的条码类型范围中读取被解码的条码类型，

* 也可设置成不规定条码类型即自动识别，但会增加解码时间，甚至出现误读的情况。

* 可读取多个不同类型的条码，并且多个条码用不同颜色的区域框区分，且读取出来的信息也以

* 相应的颜色做区分

```
dev_close_window ()
```

```
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
```

*先关闭活动图形窗口，再打开这个窗口，标识符为 WindowHandle；

*相对于界面左上角第 0 行、第 0 列，大小为 300x300 像素，颜色为黑色。

```
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0, -1, AcqHandle)
```

* 打开帧接收器（图像采集设备，如摄像头，工业相机等），参数（Parameter）详见这个算子

```
grab_image_start (AcqHandle, -1)
```

```
while (true)
```

```
grab_image_async (Image, AcqHandle, -1)
```

*开始条形码识别

```
create_bar_code_model ([], [], BarCodeHandle)
```

*必备的创建条码解码的开头，下面有一段结束代码

```
dev_update_var ('off')
```

```
dev_update_pc ('off')
```

```
dev_update_window ('off')
```

* 刷新窗体

```
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
```

```
dev_set_draw ('margin')
```

```
dev_set_line_width (3)
```

```
Colors := ['forest green','magenta','blue','red','yellow']
```

* 设置区域框的属性，如无填充色、边框线大小为 3，颜色。

```
*CodeTypes := ['auto']
```

```
CodeTypes := ['EAN-13','Code 39','EAN-8','Code 128','Code 93']
```

* 设置条码的类型，如 EAN-13 等 5 种一维码类型，或者 auto 自动获取（但解码时间较长且有误读的可能）

```

get_image_size (Image, Width, Height)
dev_display (Image)
*获取图像大小，并显示图像

count_seconds (Start)

find_bar_code (Image, SymbolRegions, BarCodeHandle, CodeTypes, DecodedDataStrings)
count_seconds (Stop)

Duration := (Stop - Start) * 1000

*开始计时、寻找并解码条形码（参数解释详见这个算子），结束计时，并计算解码的时间

dev_display (SymbolRegions)

get_bar_code_object (BarCodeObjects, BarCodeHandle, 'all', 'symbol_regions')
get_bar_code_result (BarCodeHandle, 'all', 'decoded_types', DecodedDataTypes)

*area_center (SymbolRegions, Area, Rows, Columns)

for J := 0 to |DecodedDataStrings|-1 by 1

dev_set_color (Colors[J])

select_obj (BarCodeObjects, ObjectSelected, J+1)

dev_display (ObjectSelected)

endfor

*循环获取，并且根据获取的条码数量，设置区域框的颜色

if (|DecodedDataStrings| >= 1)

*disp_message (WindowHandle, ['Found bar code(s) in ' + Duration$'3.0f' + 'ms:' + '\n Type: ' + DecodedData
Types + '\n Data: ' + DecodedDataStrings], 'window', 5*12, 12, 'black', 'true')

disp_message (WindowHandle, '读取条形码,用时{' + Duration$'3.0f' + 'ms}:' + '[' + DecodedDataTypes + ']' +
'=' + "" + DecodedDataStrings + "", 'window', 12, 12, Colors, 'true')

disp_continue_message (WindowHandle, 'black', 'true')

stop()

endif

*判断读取到数据的数量，当大于 1（数字可改为 2、3.....）

个条码被解码时，显示信息在窗体的第 12 行，第 12 列的位置，

并按顺序用不同的颜色标注

clear_bar_code_model (BarCodeHandle)

*关闭条形码解码

endwhile

close_framegrabber (AcqHandle)

*关闭帧接收器（摄像头）

```



一个摄像头解码二维码的实例及分步骤注解

```
* 2D Code generated by Image Acquisition 01
```

```
* QR Code
```

```
dev_close_window ()
```

```
dev_open_window (0, 0, 400, 400, 'black', WindowHandle)
```

*先关闭活动图形窗口，再打开这个窗口，标识符为 WindowHandle;

*相对于界面左上角第 0 行、第 0 列，大小为 400×400 像素，颜色为黑色。

```
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', 'Gsou USB2.0 Camera', 0, -1, AcqHandle)
```

* 打开帧接收器（图像采集设备，如摄像头，工业相机等），参数（Parameter）详见这个算子

* 注意摄像头的名称,可以用工具栏中的“助手”——打开新的 Image Acquisition 获取摄像头及插入代码

```
grab_image_start (AcqHandle, -1)
```

```
while (true)
```

```
grab_image_async (Image, AcqHandle, -1)
```

```
create_data_code_2d_model ('QR Code', [], [], DataCodeHandle)
```

* 二维码的创建开头的算子，clear 为结束清除的算子，见下。

```
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
```

```
dev_set_color ('forest green')
```

```
dev_set_draw ('margin')
```

```
dev_set_line_width (3)
```

```
set_data_code_2d_param (DataCodeHandle, 'default_parameters', 'enhanced_recognition')
```

* 设置选定参数的二维数据模型，参数详见这个算子

```
find_data_code_2d (Image, SymbolXLDs, DataCodeHandle, [], [], ResultHandles, DecodedDataStrings)
```

* 检测和读取二维代码符号，也支持读取二维数据模型的序列，参数详见这个算子

```
for i := 0 to |ResultHandles| - 1 by 1
```

```
select_obj (SymbolXLDs, SymbolXLD, i+1)
```

```
get_contour_xld (SymbolXLD, Row, Col)
```

```
get_string_extents (WindowHandle, DecodedDataStrings, Ascent, Descent, TxtWidth, TxtHeight)
```

```
disp_message (WindowHandle, DecodedDataStrings, 'image', max(Row-50), max([min(Col+30)-TxtWidth/2, 1]),  
'black', 'true')
```

```
endfor
```

*这段 for 循环语句的目的是让解码到的字符串（二维码的内容）显示到二维码深绿色（forest green 上面定义）的解码区域框的行列位置。

```

*disp_message (WindowHandle, DecodedDataStrings, 'window', 12, 12, 'black', 'true')

*如果不需要设置显示到区域框中间的位置，而是显示到窗体的上方或其他位置，那么不需要上面那段 for 语句，只需
这段信息显示的语句即可显示到窗体相应位置。

if (|DecodedDataStrings|>0)

disp_continue_message (WindowHandle, 'black', 'true')

stop()

endif

* if 语句，当解码一个（大于 0，可设置多个）二维码就暂停摄像头获取图像，直至按 F5.

clear_data_code_2d_model (DataCodeHandle)

endwhile

close_framegrabber (AcqHandle)

```



二维码的网址: <http://ma.m.taobao.com/buvjx>

二维码的网址: <http://jiankangjiaju.taobao.com>

代码复制粘贴即可测试。

一段可视化的条码扫描线的实例和注解

一段可视化的条码扫描线，用于辨别条码，对判定条码本身是否正确有一定作用。

* This program demonstrates visualization of bar code scanlines.

* 这个程序演示了可视化的条码扫描线。

* This visualization can be used to inspect the quality of images, where

* occlusions or print defects might prevent the bar code reader from

* successfully decoding the underlying bar code.

* 这种可视化可用于检查图像的质量，其中封闭遮断或打印缺陷可能会阻止条形码阅读器成功解码潜在的条码。

* With 'scanlines_all' all scanlines that the bar code reader would

* eventually use to decode a candidate bar code (here, colored in red).

* 'scanlines_all'用于显示标识最终的解码潜在条码的扫面线（红色部分）

* With 'scanlines_valid' are visualized all scanlines that can be decoded

* as well (here, colored in green)

* 'scanlines_valid'用于显示标识有效解码的扫描线

* 'candidate_regions'用于获取条码潜在区域

```
dev_close_window ()
```

```
dev_open_window (0, 0, 500, 500, 'black', WindowHandle)
```

```
set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
```

```
dev_set_draw ('margin')
```

```
dev_update_window ('off')
```

```
dev_set_line_width (2)
```

```
create_bar_code_model ([], [], BarCodeHandle)
```

```
set_bar_code_param (BarCodeHandle, 'element_size_min', 1.5)
```

*这个参数是指条码的条（黑线）和空（空白间隔）的最小宽度，对于尺寸很大的条码，这个最小值 element_size_min

*要相应增加，比如一个较大的条码图片，用 1.5 可能就读不到，因为条和空的宽度很大，用 2 或 2.5 以上就可以解码，

*但数值越大，解码的时间就越长。还有一个 element_size_max 的参数，和这个相反，属于限制最大宽度，宽度不能太大，

*不然相邻的条和空就会被识别成一个了。一般这两个可以一起用

```
set_bar_code_param (BarCodeHandle, 'persistence', 1)
```

* persistence 这个参数没有查到，暂未知

```
read_image (Image,'L:/Halcon test/barcode/test_020.jpg')
```

```
get_image_size (Image, Width, Height)
```

```
dev_set_window_extents (0, 0, Width, Height)
dev_display (Image)
find_bar_code (Image, SymbolRegions, BarCodeHandle, 'EAN-13', DecodedDataStrings)
* 寻找条码，并获取解码值。
get_bar_code_object (Candidates, BarCodeHandle, 'all', 'candidate_regions')
dev_set_color ('magenta')
dev_display (Candidates)
* 获取条码的扫描线候选区域。
get_bar_code_object (AllScanlines, BarCodeHandle, 'all', 'scanlines_all')
dev_set_color ('red')
dev_display (AllScanlines)
* 获取所有的扫描线（扫描线应该是指在一定的高度内读取到的条和空的边缘形成的线），这个读取不论是否能解码，均读取显示。
get_bar_code_object (ValidScanlines, BarCodeHandle, 'all', 'scanlines_valid')
dev_set_color ('green')
dev_display (ValidScanlines)
* 获取有效的条码扫描线
if (|DecodedDataStrings|>0)
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
endif
* if 语句，当读到了数值就停下来。
clear_bar_code_model (BarCodeHandle)
```

下图是“简单活”提供的原图分析的结果：

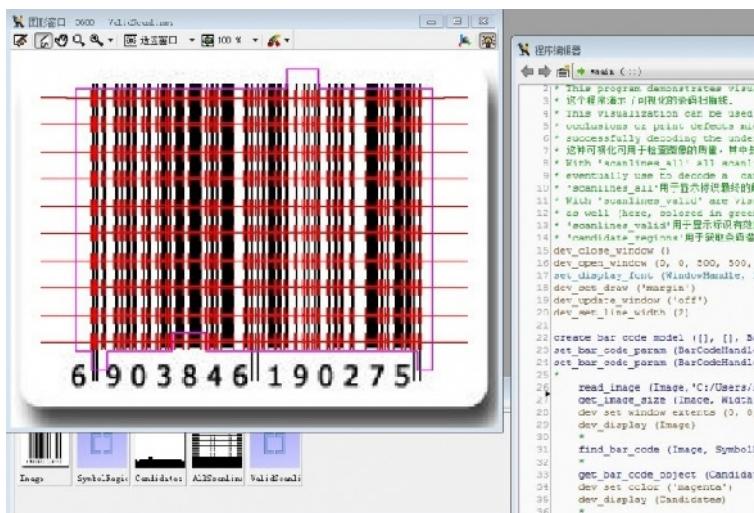


图 1: 一个可能错误的 EAN-13 条码。

下图是在群里和一位正在做一维码项目的朋友讨论过程的一个分析：

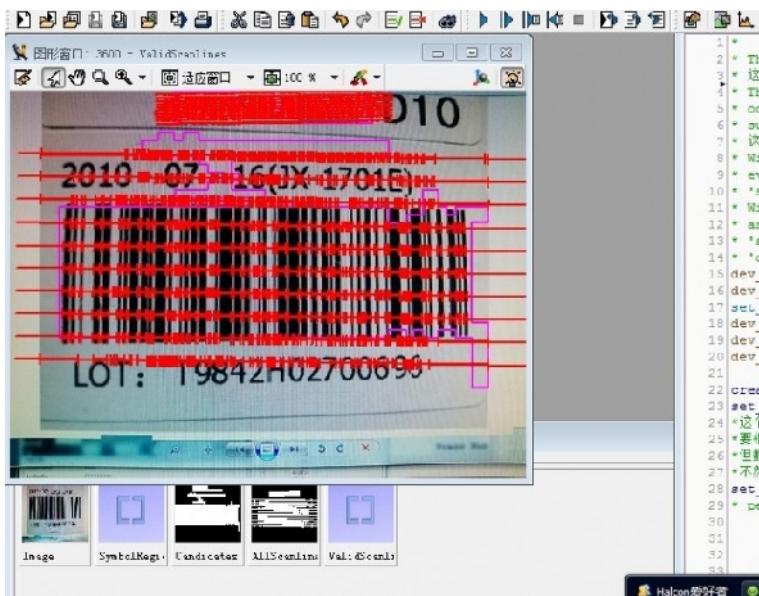


图 2：她提供的原图分析

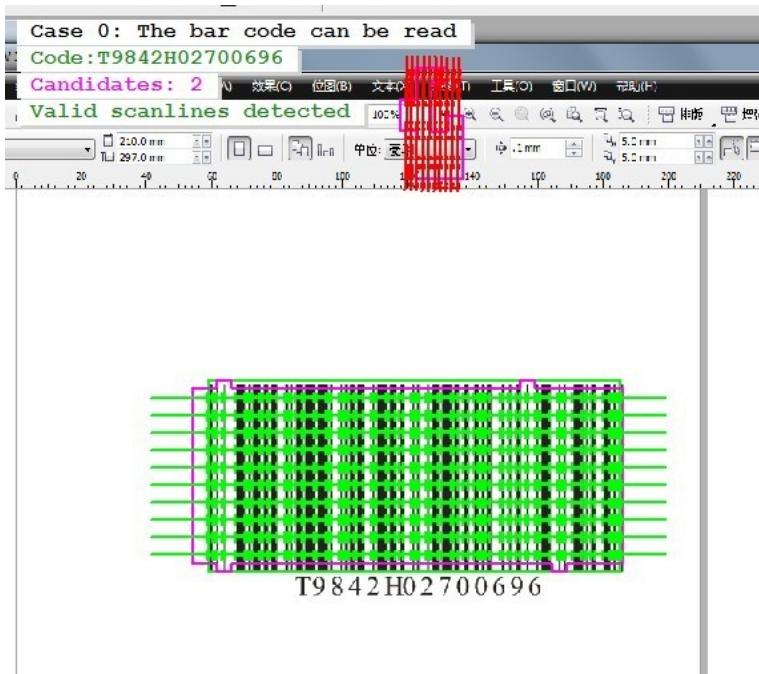


图 3：我用 CorelDraw 做的条码值一样的 Code128 条形码

该项目有可能是条码制作时有偏差或错误，或者是工业相机采集时的问题，或其他未知的，还在讨论中。

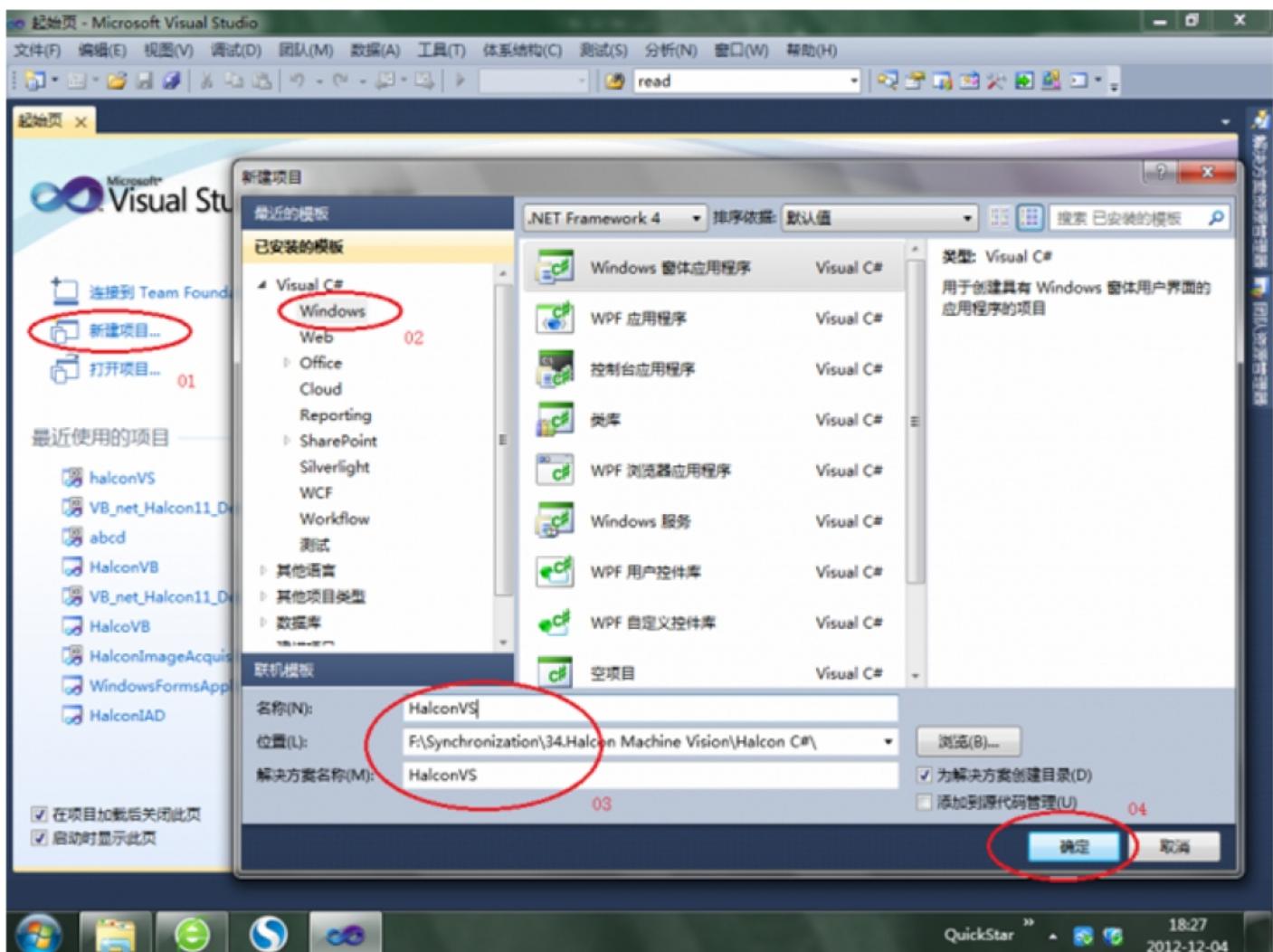
Visual Studio 10.0 设置引用 HalconDotNet.dll

开始做 Halcon 的上位机，选用 Visual Studio 10.0 平台。

具体设置以 C#，VB 为例子，如下：

C#

第一步：



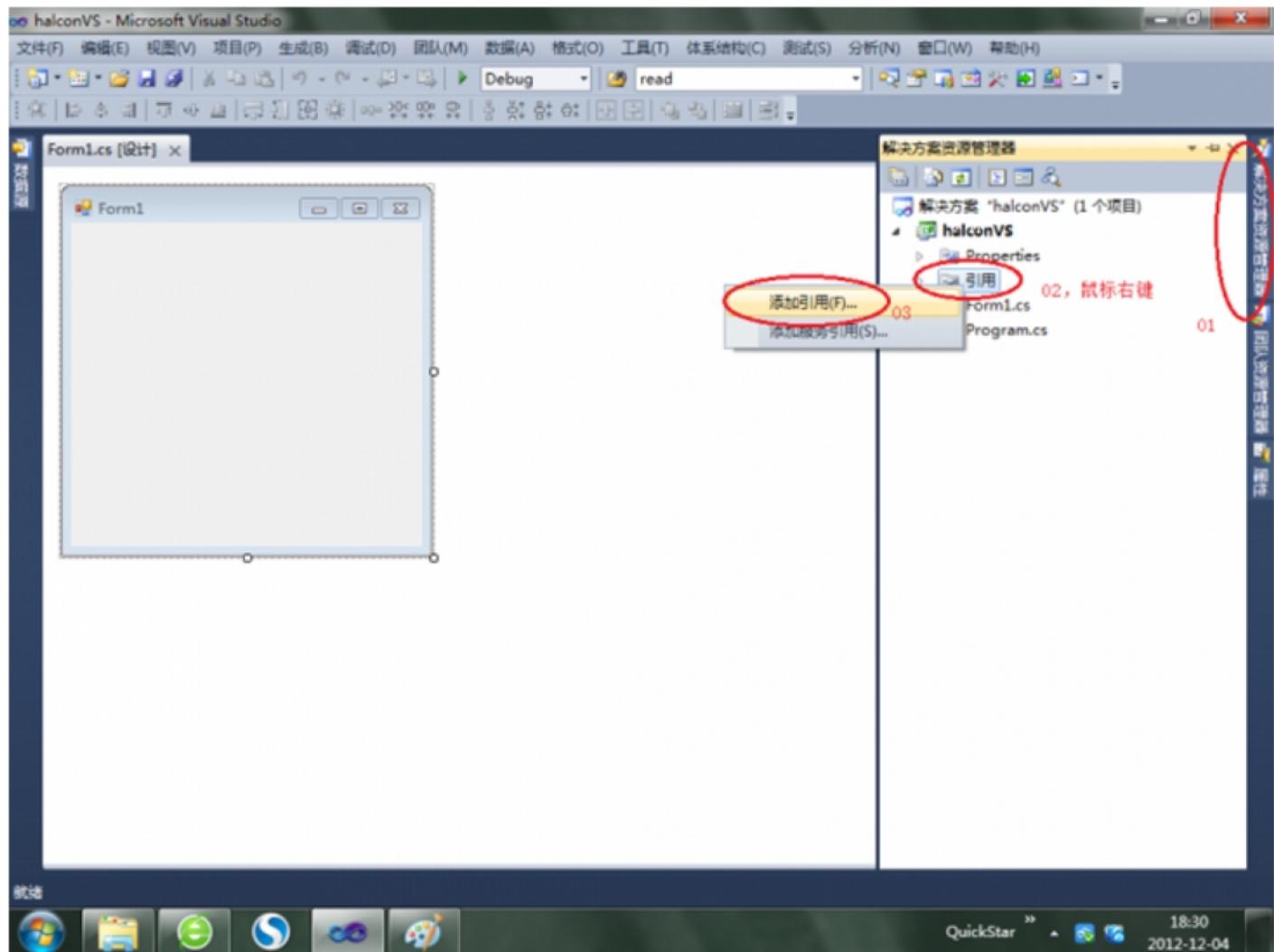
01.新建项目

02.选择 Visual C#

03.修改文件名

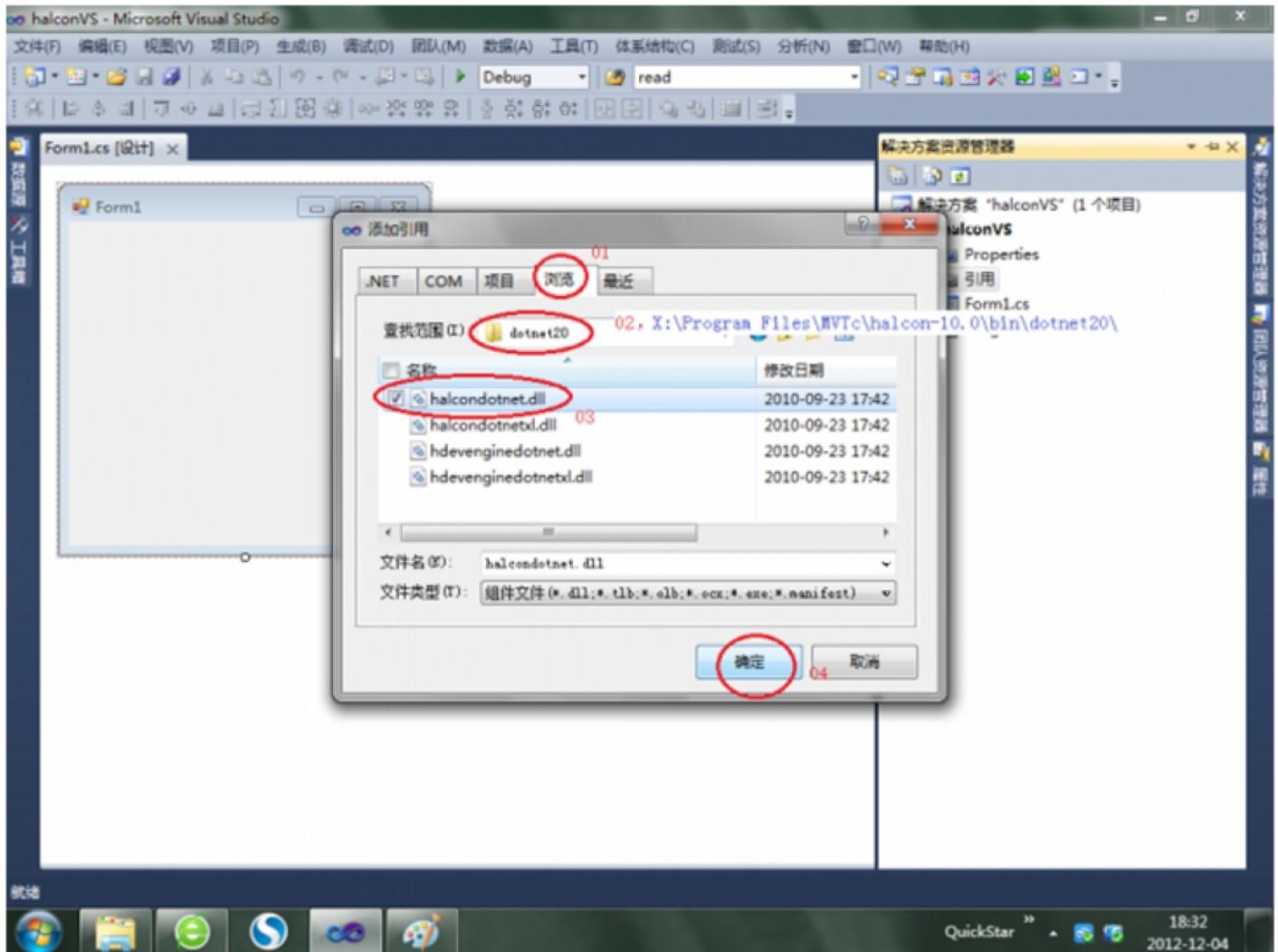
04.确认

第二步



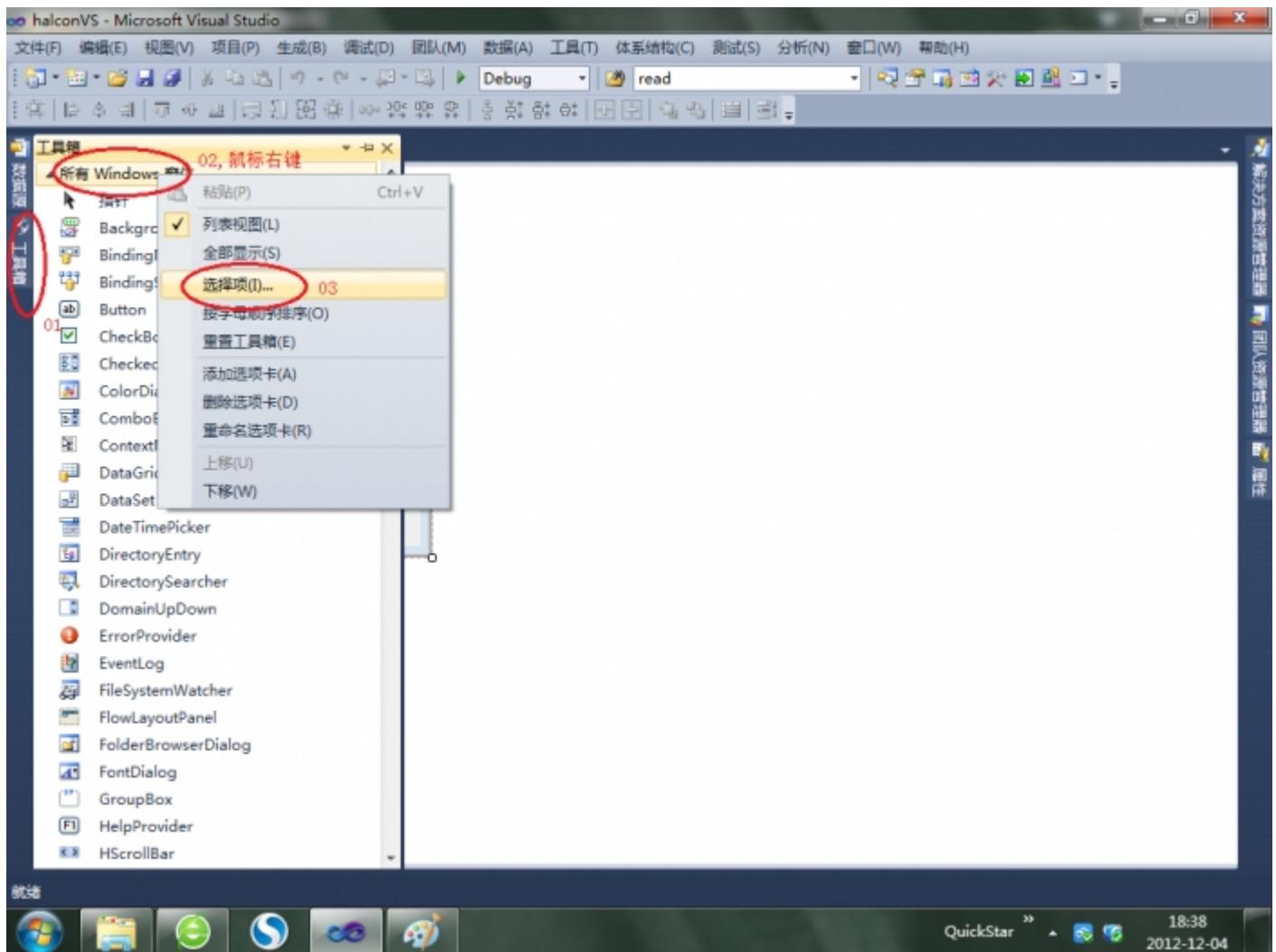
- 01.选择“解决方案资源管理器”
- 02.选择“引用”，右键
- 03.选择“添加引用”

第三步



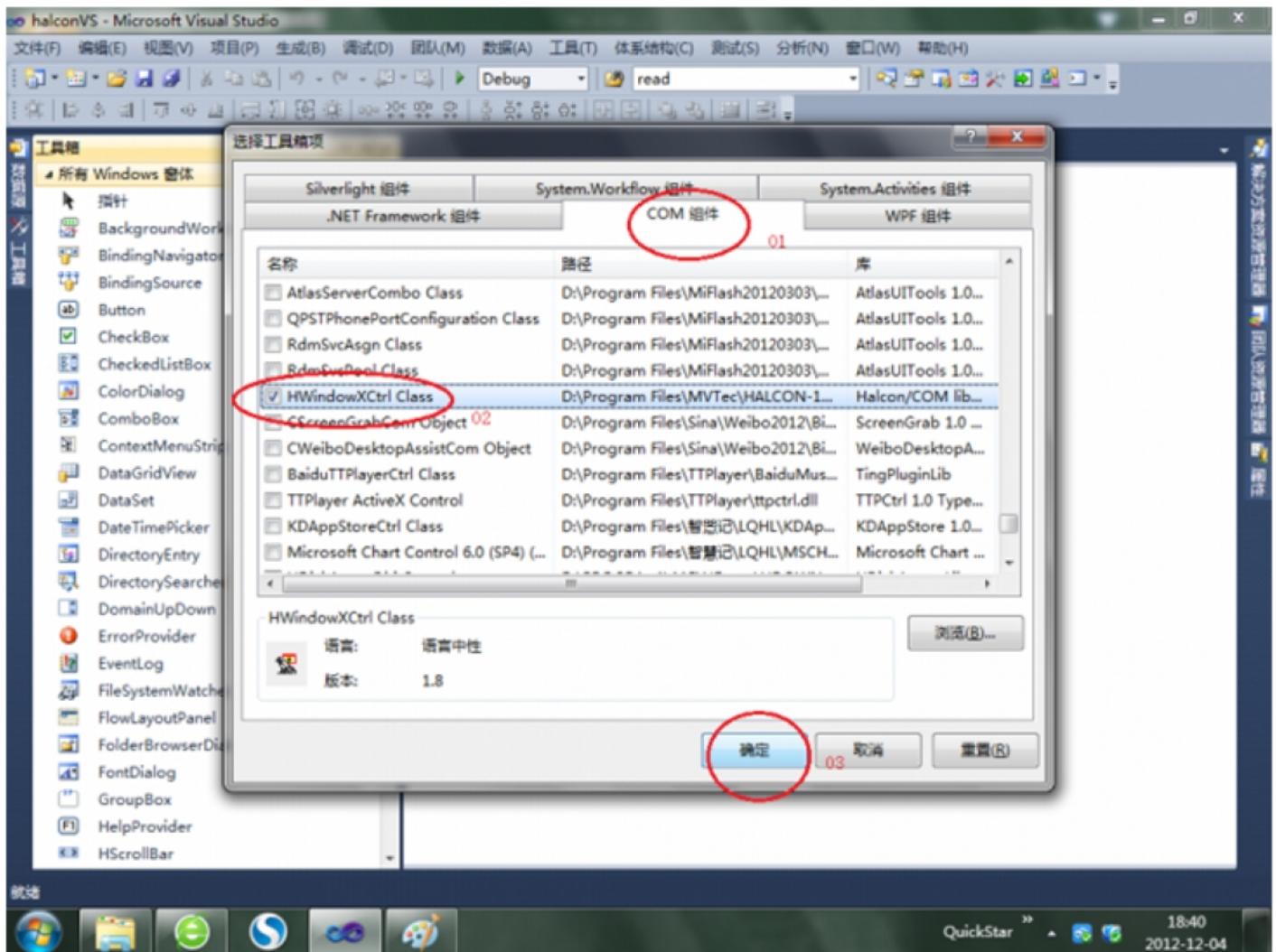
- 01.在添加引用选项卡选择浏览
- 02.选择路径: X:\Program Files\MVTec\HALCON-10.0\bin\dotnet20
- 03.选择“HalconDotNet.dll”
- 04.确认

第四步

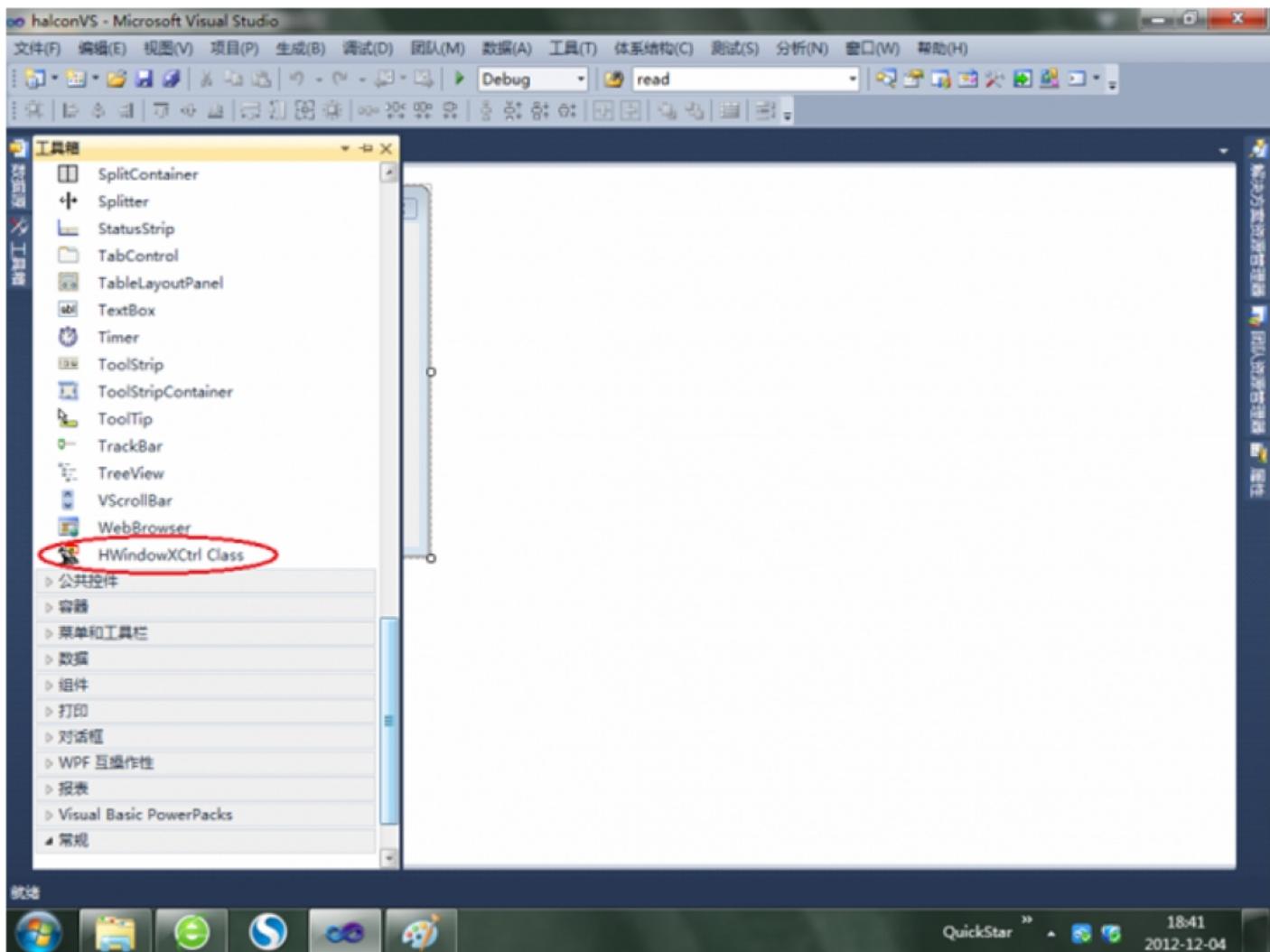


- 01.选择“工具箱”
- 02.选择“所有工具”或任意工具控件，右键
- 03.选择“选择项”

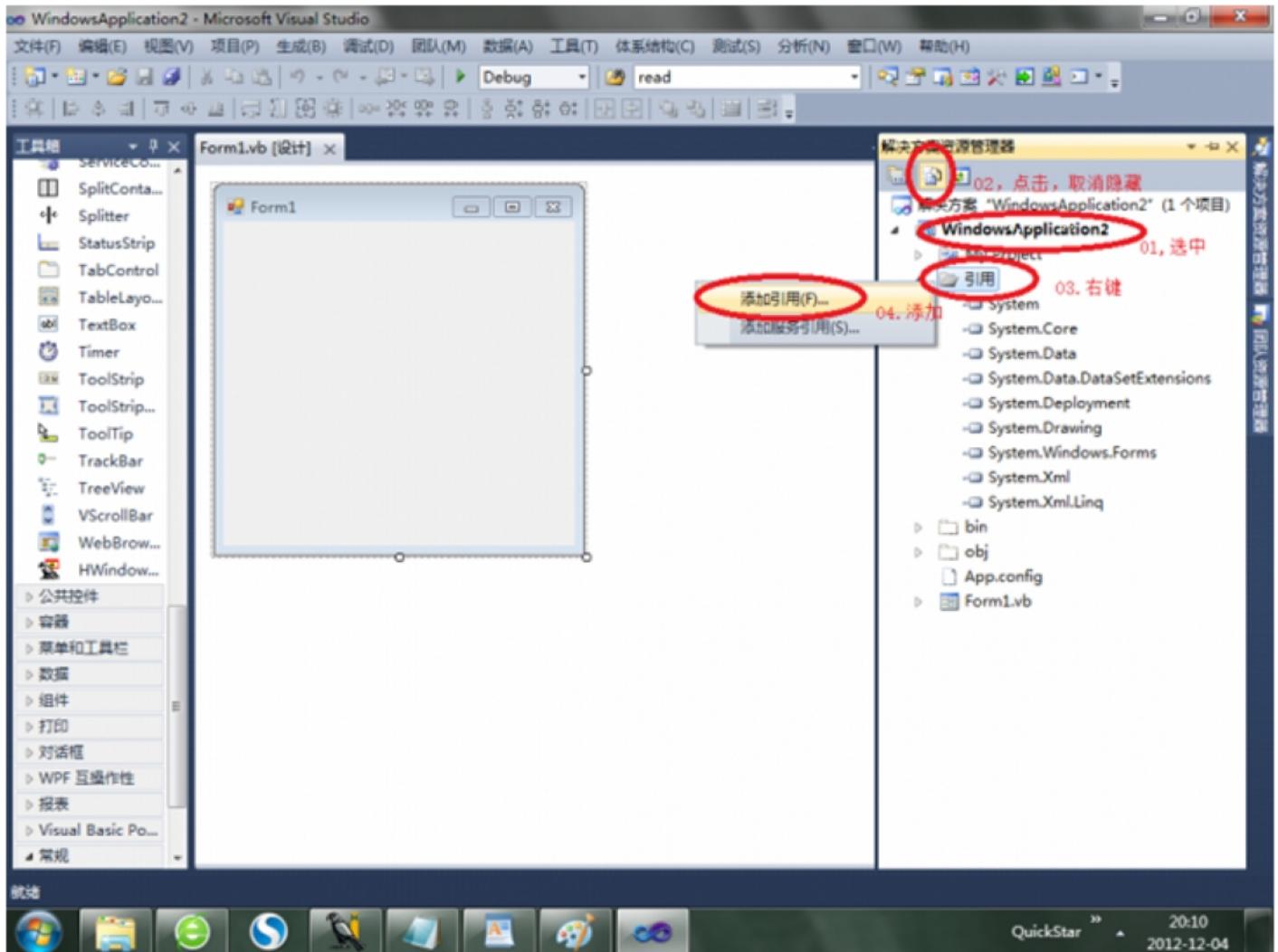
第五步



选择 Halcon 图像控件



还有 VB 的设置，如下图：

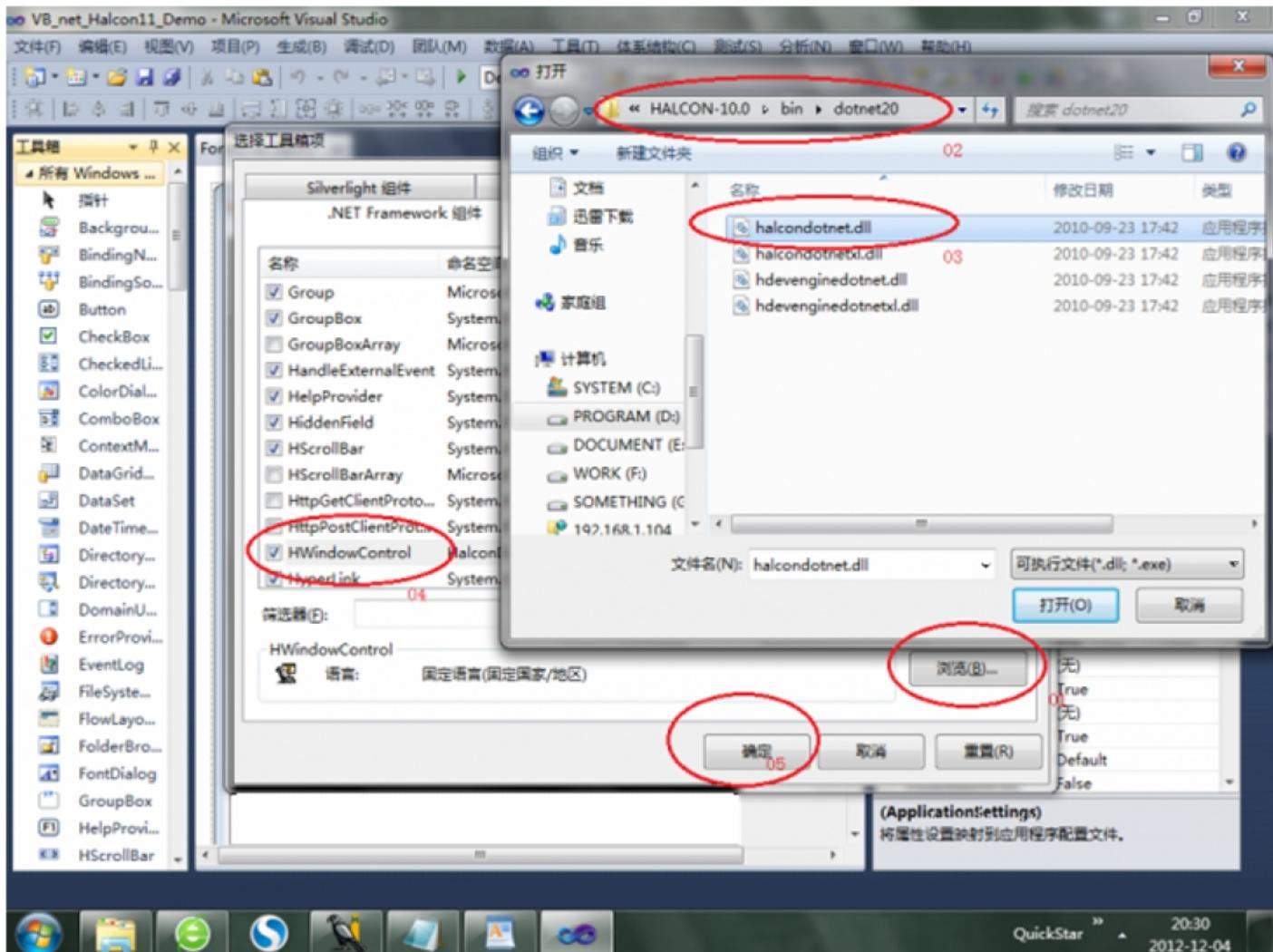


01.选择项目名称

02.点击显示全部文件，取消隐藏，就可以看到引用

03.引用，邮件

04.添加，之后的步骤和 C#一样



选择工具控件，步骤和 C# 基本一样。

01. 选择“工具箱”
02. 选择“所有工具”或任意工具控件，右键
03. 选择“选择项”
04. 在“.net framework 组件”的选项卡中，选择浏览
05. 出来的对话框中，选择 X:\Program Files\MVTec\HALCON-10.0\bin\dotnet20
06. 选择“HalconDotNet.dll”
07. 确认
08. 这时候的工具栏的控件是 HWinwow 控件。

kamuixiao:

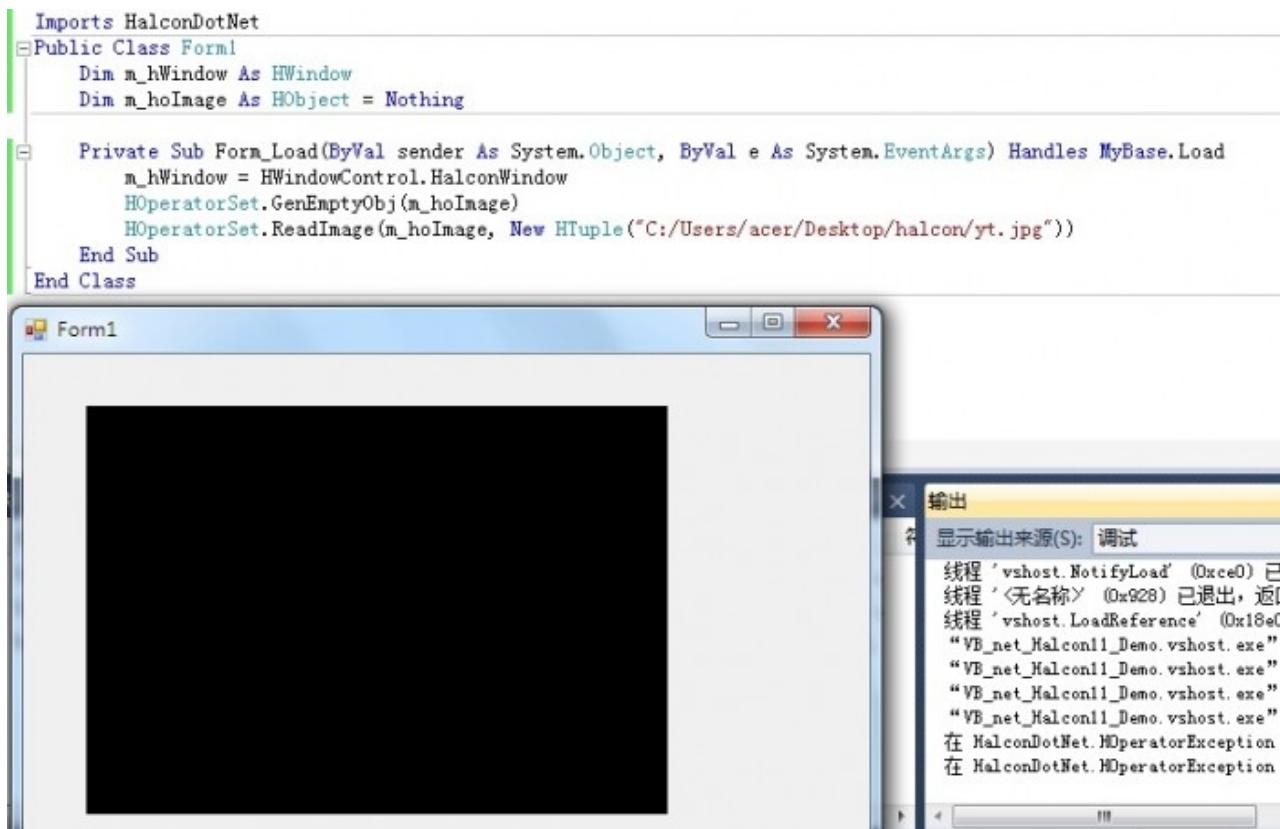
X:\Program Files\MVTec\HALCON-10.0\bin\dotnet20

路径下的无法使用，在.net framework 组件无法找到

X:\Program Files\MVTec\HALCON-10.0\bin\dotnet35

路径下的可用，在.net framework 组件下可以找到，并且在工具箱上能够找到并使用，但是功能无法实现，如下图我不作其他动作，程序代码内容仅是运行时候读取一张图片，

在调试的时候能够运行，但读不到图片……楼主在 VB.NET 环境下正常运行了吗？求解~



x8515618:

定义错了

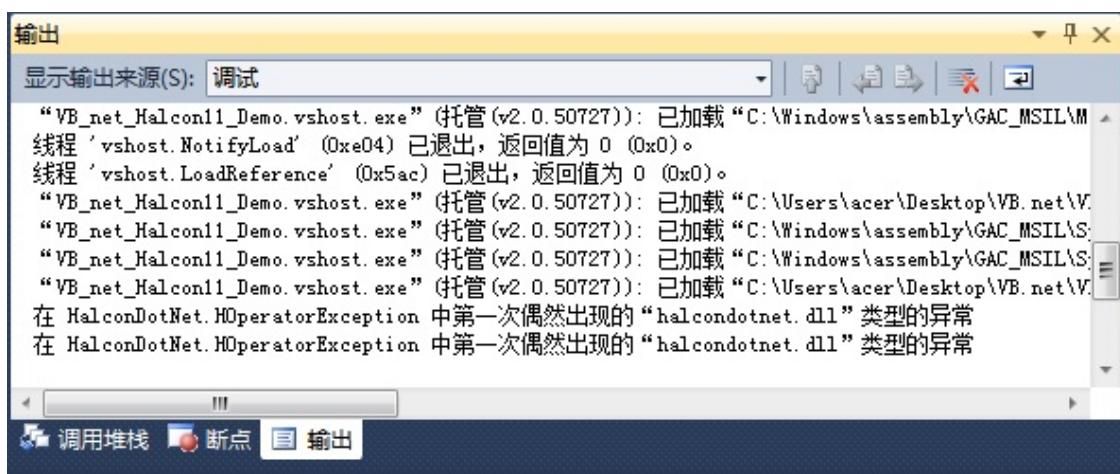
dim m_hWindow As Htuple '这是一个句柄，不是窗体控件

x8515618:

Dim m_hWindow As Htuple = Nothing

kamuixiao:

谢谢楼上的建议，但是我试了下发现仍然不行，我觉得问题不是出在这里，不知道使用 VB.NET 的各位有没有遇到过，反正我是没能正常实现过一次的，输出显示的错误情况如图：



x8515618:

你只是读取了图片，都没有显示图片呢

灵峤 lencue:

你最开始新建项目的时候选择的“.net 框架”版本是多少的？.net 2.0 就用 dotnet20 文件夹，3.5 就用 dontnet35 文件夹，你再查看一下，会不会是这个问题，看能不能解决。

irun999:

请问楼主，用在 VS2008 中用 C# 读入一直比较大的图片的时候只是显示部分图片，如何解决？

灵峤 lencue:

```
dev_resize_window_fit_image(ho_Image, 0, 0, 300, 400);
```

用这个算子，在 C# 的 program.cs 中可以用的，后面的 300,400 就是显示这个图片的大小的，会被缩放到这个比例。

irun999:

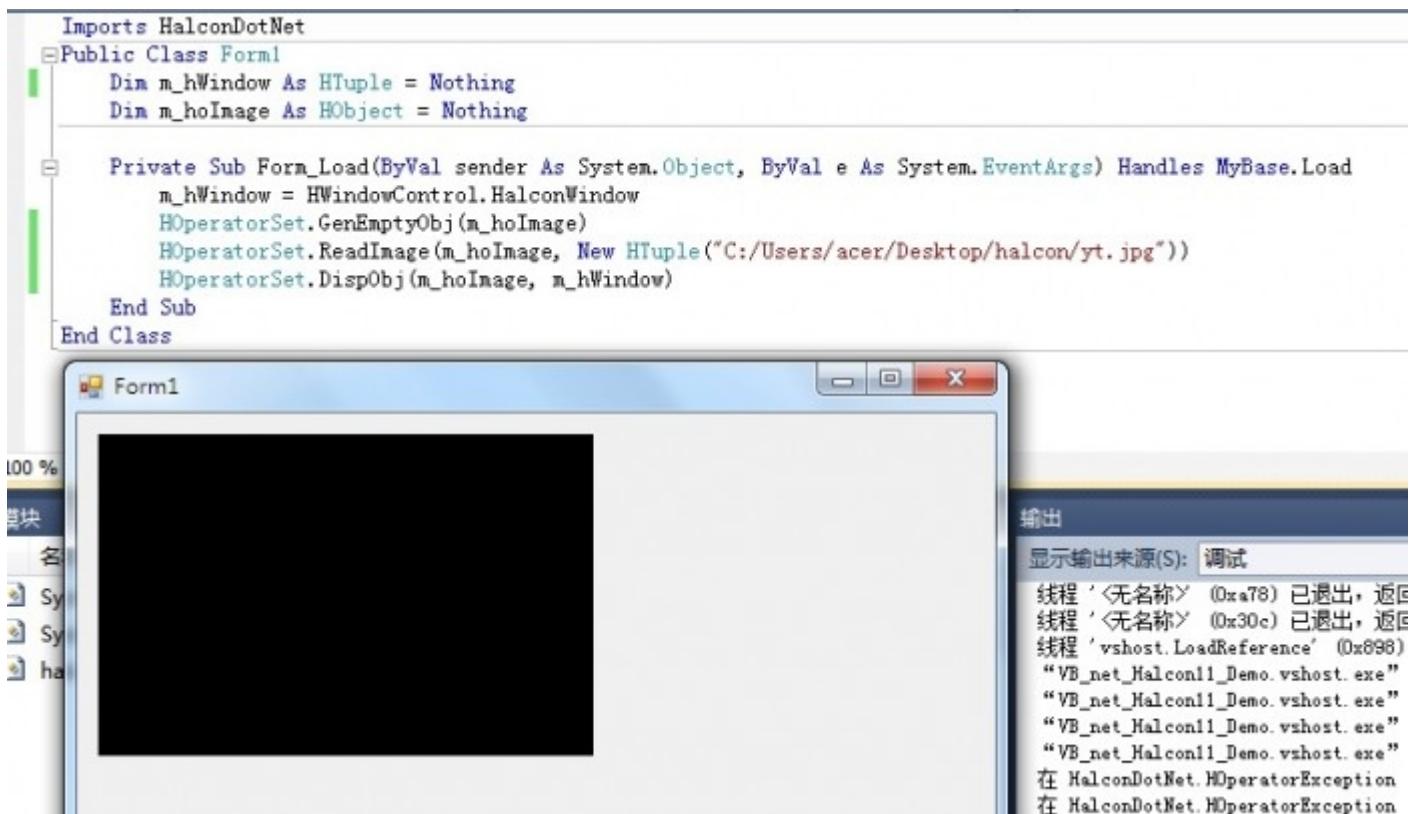
请问在 VS 2005 C# 处理视频，程序老是死掉，怎么办？视频无法实时显示。是不是视频太大了，VS2005 处理不够来？该怎么办？

灵峤 lencue:

采用 C# 的多线程，具体怎么使用，我还在学习中，要是有进展，再一起讨论。

kamuixiao:

好吧，我写漏了，但事实上我把显示补上，然后按照兄台所说修改定义后仍然没能成功！~附图：



irun999:

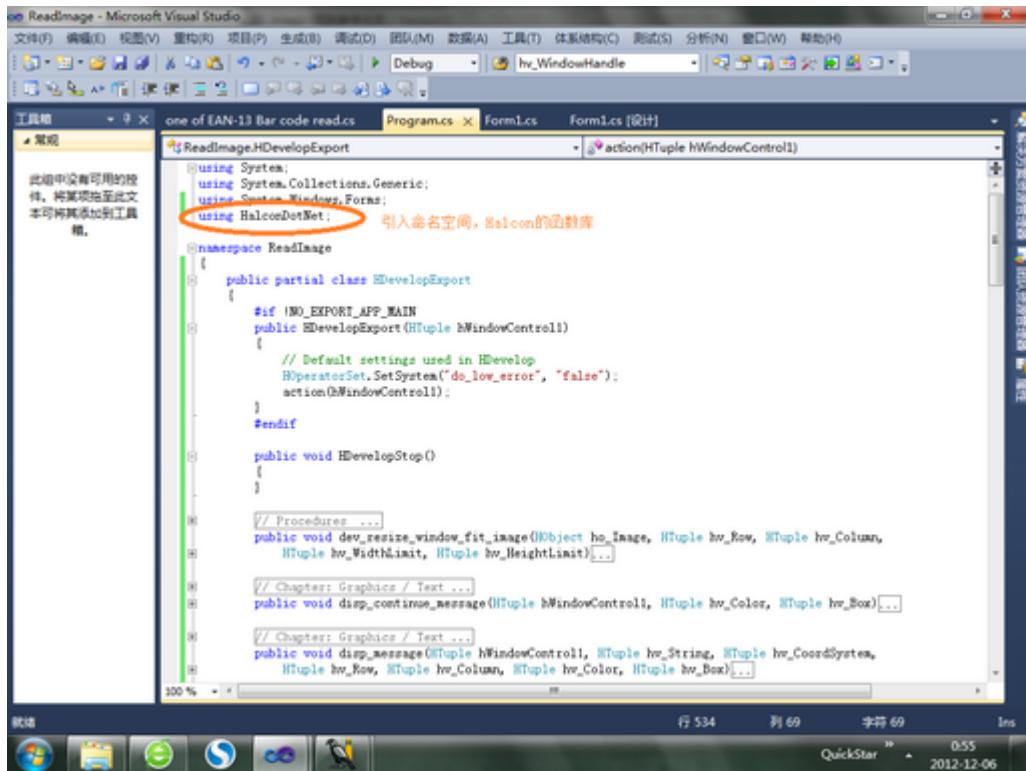
今天下午发现视频无法显示是因为 hdevelop 导出程序的死循环函数里没有 HOperatorSet.DispObj 显示函数。但是关闭程序就会出现 vhost.exe 无响应。这个问题可能得多线程解决了。

C#导出的代码，怎么调用？

C#导出的代码，怎么调用？

下面抛砖引玉，我自己的做法，可能不是很正确，请群友指正。

01.用 using HalconDotNet; 引用命名空间，调用 Halcon 函数库



```
ReadImage.HDevelopExport
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using HalconDotNet; // 引入命名空间，Halcon的函数库

namespace ReadImage
{
    public partial class HDevelopExport
    {
        #if !__MO__EXPORT_APP_MAIN
        public HDevelopExport(HTuple hWindowControl)
        {
            // Default settings used in HDevelop
            HOperatorSet.SetSystem("do_low_error", "false");
            action(hWindowControl);
        }
        #endif

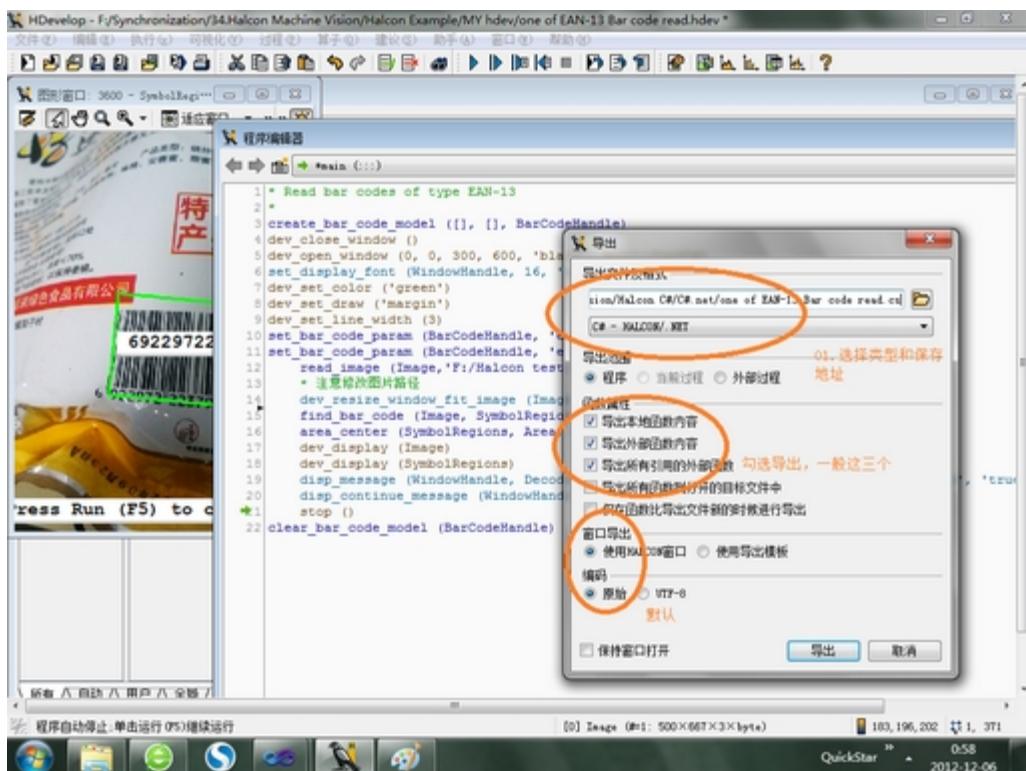
        public void HDevelopStop()
        {
        }

        // Procedures ...
        public void dev_resize_window_fit_image(HObject ho_Image, HTuple hv_Row, HTuple hv_Column,
                                                HTuple hv_WidthLimit, HTuple hv_HeightLimit){...}

        // Chapter: Graphics / Text ...
        public void disp_continue_message(HTuple hWindowControl, HTuple hv_Color, HTuple hv_Box){...}

        // Chapter: Graphics / Text ...
        public void disp_message(HTuple hWindowControl, HTuple hv_String, HTuple hv_CoordSystem,
                               HTuple hv_Row, HTuple hv_Column, HTuple hv_Color, HTuple hv_Box){...}
    }
}
```

02.在 Halcon-HDevelop 里导出代码，设置如下：导出类型为 C#.net



03. 打开刚导出的代码，复制 HDevelopExport。

```
// 导出的C#代码
// File generated by HDevelop for HALCON/DOTNET (C#) Version 10.0

using System;
public partial class HDevelopExport
{
    [STAThread]
    static void Main(string[] args)
    {
        new HDevelopExport();
    }
}
```

04. 打开 Program.cs，复制代码

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using HalconDotNet;

namespace ReadImage
{
    public partial class HDevelopExport
    {
        static class Program
        {
            /// <summary>
            /// 应用程序的主入口点。
            /// </summary>
            [STAThread]
            static void Main()
            {
                Application.EnableVisualStyles();
                Application.SetCompatibleTextRenderingDefault(false);
                Application.Run(new Form1());
            }
        }
    }
}
```

05. 声明 HDevelopExport、action 的元组

The screenshot shows the Microsoft Visual Studio IDE with the code editor open. The file is named 'one of EAN-13 Bar code read.cs'. The code defines a class 'ReadImage.HDevelopExport' with two methods: 'HDevelopStop' and 'action'. The 'action' method is highlighted with a red box. A note in the code states: 'hWindowControl1是在form1.cs[设计]中建立的Halcon控件的名称，在其属性中可看到。' (The name of the Halcon control in form1.cs [design] is hWindowControl1, which can be seen in its properties.)

```
#if !MO_EXPORT_APP_WIN
public HDevelopExport(HTuple hWindowControl1)
{
    // Default settings used in HDevelop
    OperatorsSet_SetSystem("do_low_error", "false");
    action(hWindowControl1);
}
#endif

public void HDevelopStop()
{
}

// Procedures ...
public void dev_resize_window_fit_image(HObject ho_Image, HTuple hv_Row, HTuple hv_Column,
    HTuple hv_WidthLimit, HTuple hv_HeightLimit)...

// Chapter: Graphics / Text ...
public void disp_continue_message(HTuple hWindowControl1, HTuple hv_Color, HTuple hv_Box)...

// Chapter: Graphics / Text ...
public void disp_message(HTuple hWindowControl1, HTuple hv_String, HTuple hv_CoordSystem,
    HTuple hv_Row, HTuple hv_Column, HTuple hv_Color, HTuple hv_Box)...

// Chapter: Graphics / Text ...
public void set_display_font(HTuple hWindowControl1, HTuple hv_Size, HTuple hv_Font,
    HTuple hv_Bold, HTuple hv_Slant)...

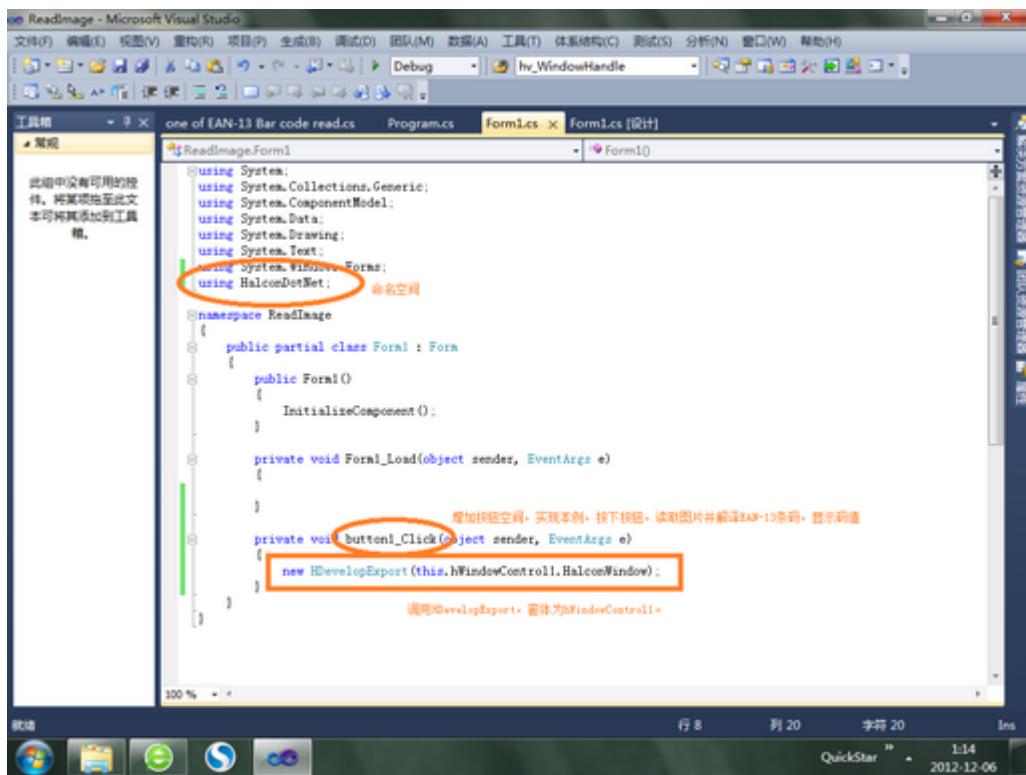
// Main procedure
private void action(HTuple hWindowControl1)
{
```

06. 建立一个窗体，更改窗体的名称

The screenshot shows the Microsoft Visual Studio IDE with the code editor open. The file is named 'one of EAN-13 Bar code read.cs'. The code implements the 'action' method from the previous screenshot. A specific line of code is highlighted with a red box: 'if (HDevWindowStack.Push(hWindowControl1);'. A note on the left margin says: '若要用HALCON建 立的显示框中显 示图片，这需要 重新操作 OpenWindow' (If you want to display an image in the display window created by HALCON, you need to re-operate OpenWindow). Another note at the bottom left says: '若要用HALCON自 动读取图片工具， 请参阅注释' (If you want to automatically read images using the HALCON tool, please refer to the comments).

```
if (HDevWindowStack.Push(hWindowControl1);
```

07. 建立一个按钮，在 form1.cs，按钮动作里添加输出 HDevelopExport。



```
ReadImage - Microsoft Visual Studio [1]
文件(F) 编辑(E) 视图(V) 重构(R) 项目(P) 生成(I) 调试(D) 团队(M) 数据(A) 工具(T) 体系结构(C) 测试(S) 分析(N) 窗口(W) 帮助(H)
... Debug hv_WindowHandle Form1.cs [设计]
工具箱 X one of EAN-13 Bar code reads.cs Program.cs Form1.cs X Form1.cs [设计]
ReadImage.Form1
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using HalconDotNet; //命名空间

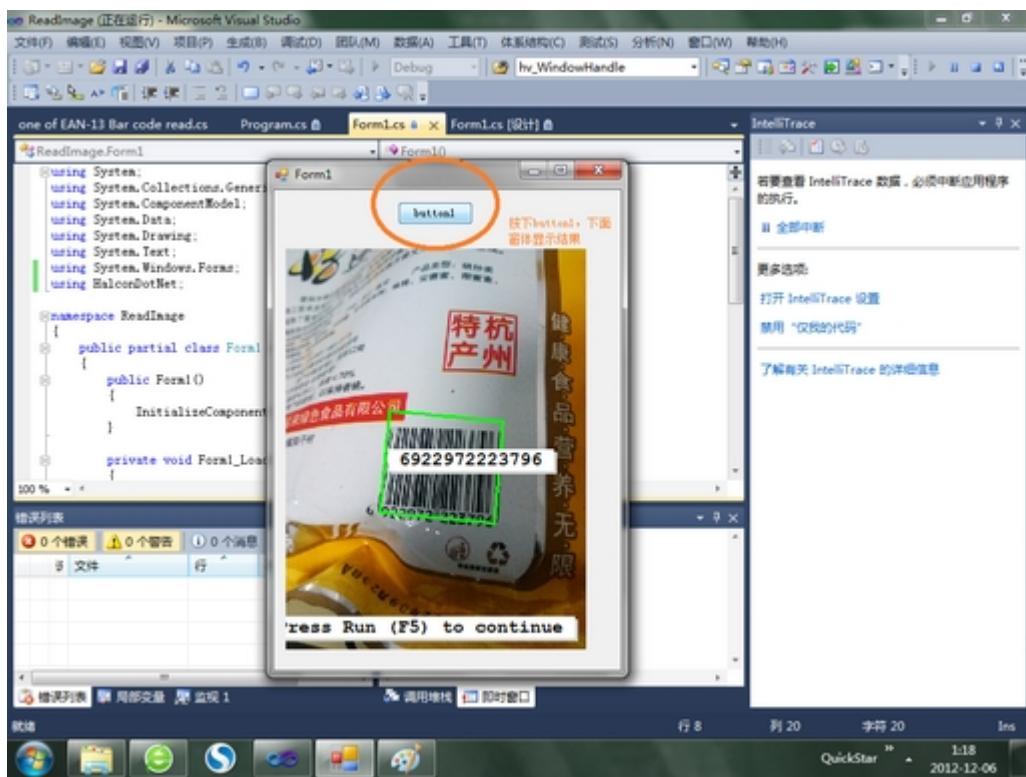
namespace ReadImage
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

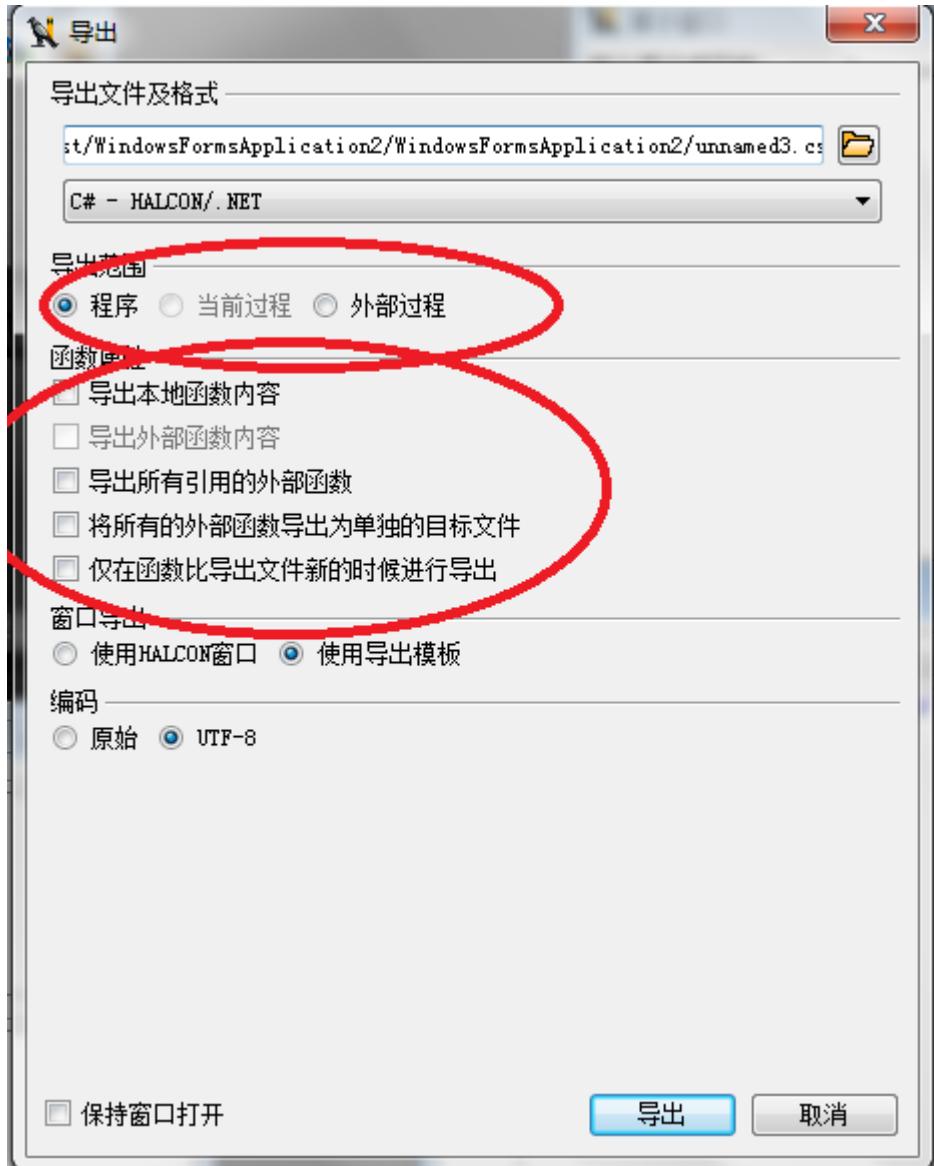
        private void button1_Click(object sender, EventArgs e)
        {
            new HDevelopExport(this, hWndControl1, HalconWindow);
        }
    }
}
```

08. 生成——运行，按下按钮，读取图片-解码。

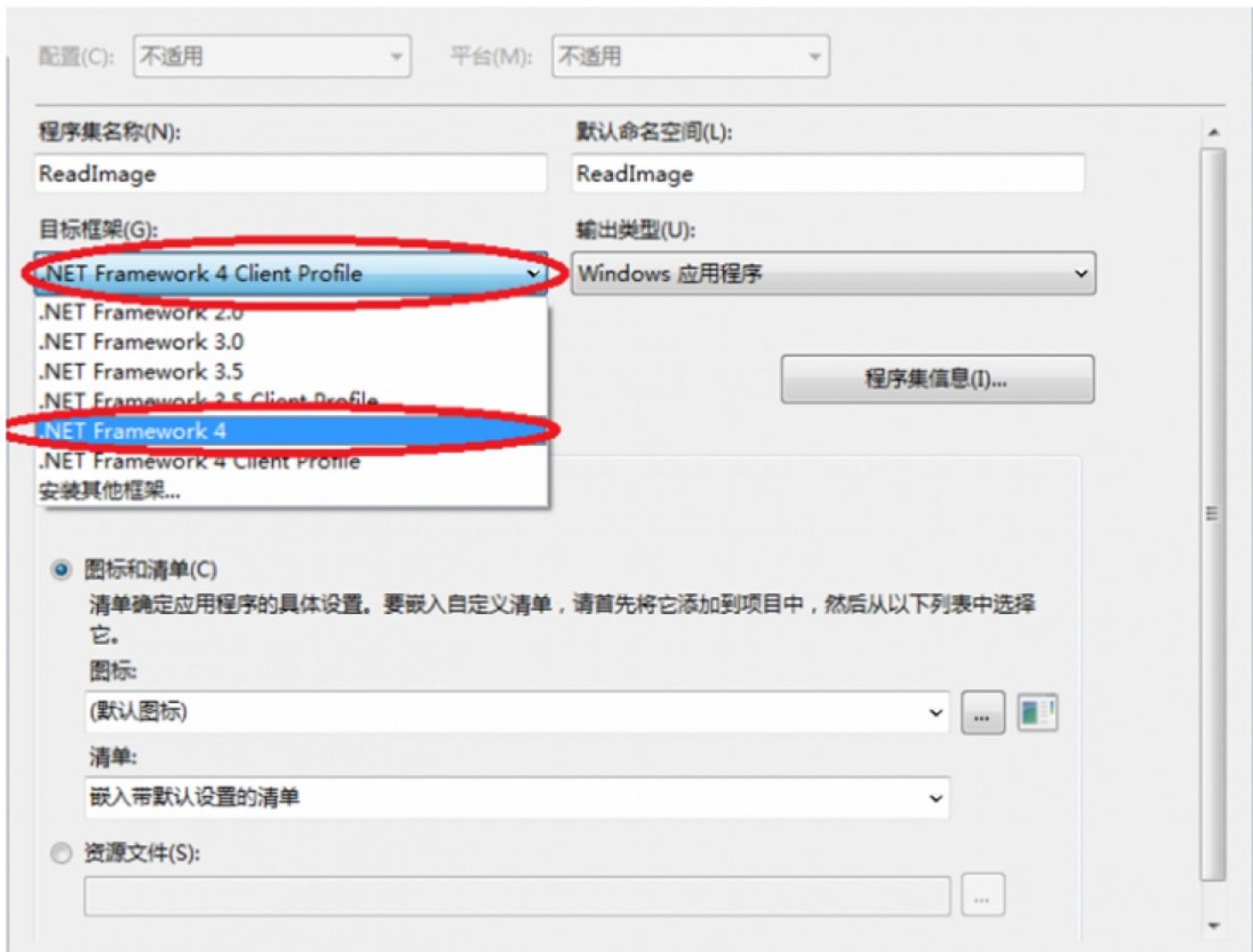
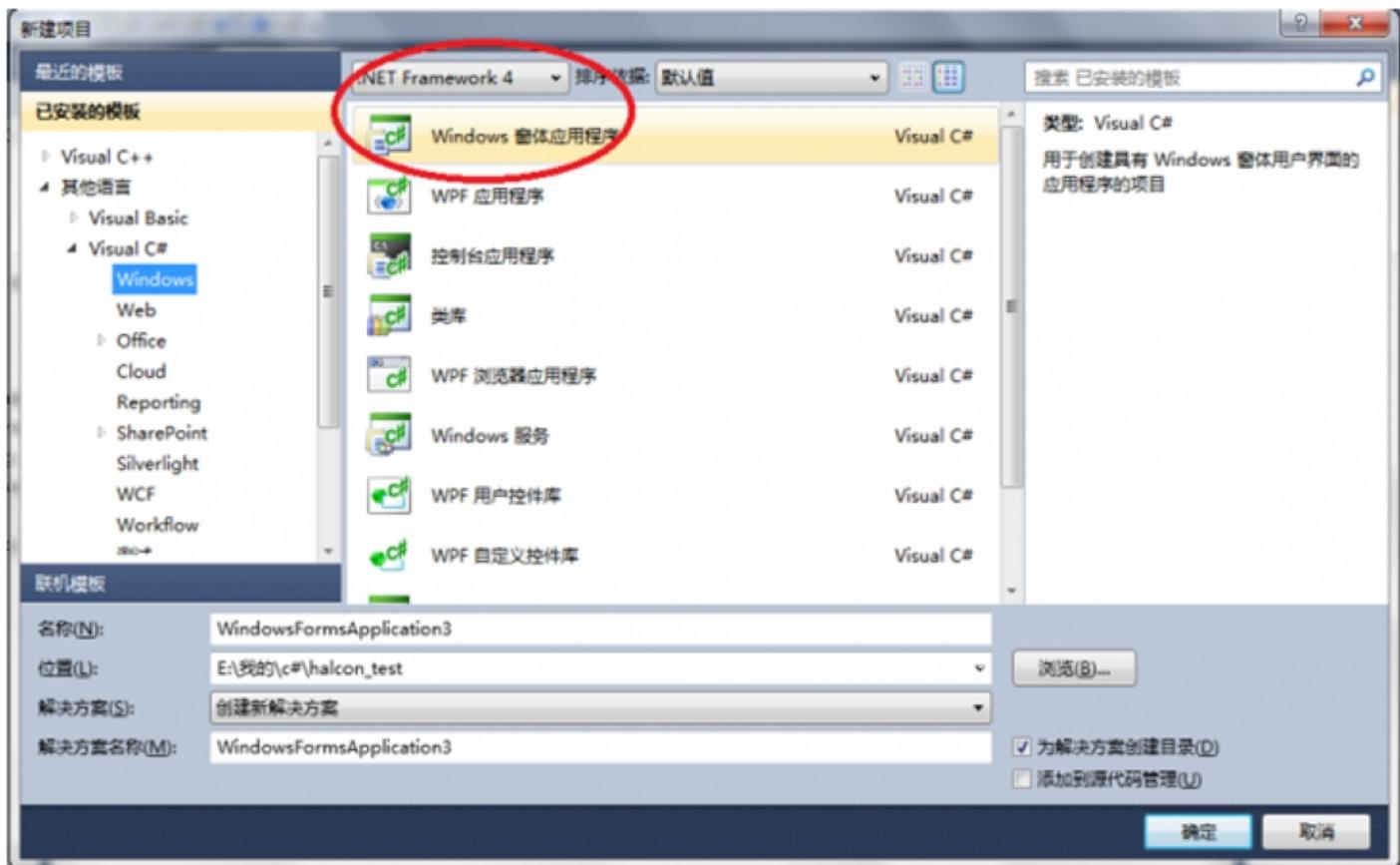


看了 灵峤 lencue 发的帖后，我来谈谈我如何在 C# 中调用 Halcon 导出的代码，学习中求指教！

首先，我在从 Halcon 中导出 C# 代码时，外部函数，我没有导出，因为它所引用的外部函数挺多了。图中没有勾选东西。（不知道那些有什么用，知道的请指导下）



然后建立 C# 工程，我在这里提醒大家的是，在建立的时候，目标框架是.NET Framework 4，但是建立完成后，在解决方案资源管理器空白处右击，选中属性时，目标框架显示的却是.NET Framework 4 Client Profile，因此我们要在下拉框中，改为.NET Framework 4，不然在执行时会报错。（这个我也不太清楚原因）



建立完成，引用“halcondotnet”，然后添加一个 button 按钮，和一个 HWindow Control 控件。至于如何添加引用和如何添加“HWindow Control 控件”。详见 [xiong023xiao](http://www.halconbbs.com/read.php?tid-178.html) 发的帖子 <http://www.halconbbs.com/read.php?tid-178.html>，虽然他最后写的那部分我没看懂，自己问人才搞定的。打开导出的代码，发现，内容不多，就一个 HDevelop Export 类，类中就三个函数，以后我们只要调用公共函数 public void RunHalcon(HTuple Window) 就可以。复制这些代码。

```
ReadImage.cs*  X  Form1.cs  Form1.cs [设计]
HDevelopExport  RunHalcon(HTuple Window)

using System;
using HalconDotNet;

public partial class HDevelopExport
{
    public HTuple hv_ExpDefaultWinHandle;

    // Main procedure
    private void action()
    {
        // Local iconic variables
        HObject ho_Image;
        // Initialize local and output iconic variables
        HOperatorSet.GenEmptyObj(out ho_Image);

        ho_Image.Dispose();
        HOperatorSet.ReadImage(out ho_Image, "fabrik");
        HOperatorSet.DispObj(ho_Image, hv_ExpDefaultWinHandle);
        ho_Image.Dispose();
    }

    public void InitHalcon()
    {
        // Default settings used in HDevelop
        HOperatorSet.SetSystem("do_low_error", "false");
    }

    public void RunHalcon(HTuple Window)
    {
        hv_ExpDefaultWinHandle = Window;
        action();
    }
}
```

双击 button 控件，打开后台代码，将复制的那个 HDevelop Export 类粘贴到 public partial class Form1 : Form 里面，不可以放在 private void button1_Click(object sender, EventArgs e) 这个函数里

```
namespace ReadImage
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            HDevelopExport HD = new HDevelopExport();
            HD.RunHalcon(hWndControl1.HalconWindow);
        }

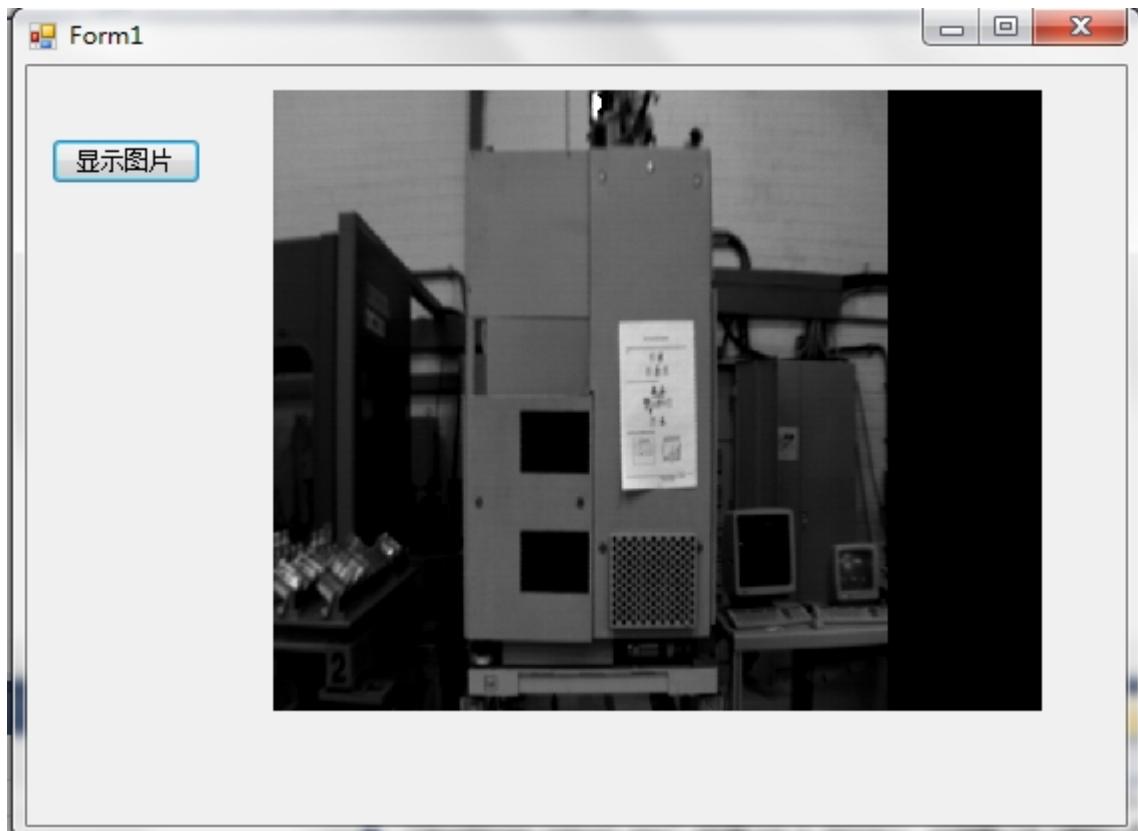
        public partial class HDevelopExport...
    }
}
```

好了，下面我们要做的就是在 `private void button1_Click(object sender, EventArgs e)` 这个事件中调用导出的函数，我们知道，导出的是个类，因此，要先产生一个类的实体，然后，调用实体中 `RunHalcon` 这个函数

```
private void button1_Click(object sender, EventArgs e)
{
    HDevelopExport HD = new HDevelopExport();
    HD.RunHalcon(hWndControl1.HalconWindow);
}
```

一切搞定，程序成功执行！

最后，贴上程序执行的结果。



灵峤 lencue:

多谢 reed1989 指教，有几点一起讨论的东西：

01. 从 Halcon 中导出 C# 代码时的函数属性（楼主的第一张图），希望有高手解析一下；
02. 引用“halcondotnet” 和如何添加引用和如何添加“HWindow Control 控件”在我的一个贴里讲到（附图）
<http://www.halconbbs.com/read.php?tid-309.html>
03. 楼主的例子中如果遇到了一些 halcon 的外部函数，那么不导出来的话，如何引用呢？

reed1989:

感谢灵峤的讨论，说实在的，我刚刚接触这方便，很多东西不懂，比如 halcon 导出的外部函数，我记得里面有些 `dev_open_window_fit...` 什么的函数，感觉挺像 halcon 自带的那些算子，而在 C# 中真正执行的就是通过公共接口 `RunHalcon()` 函数 执行私有成员函数 `action()` 函数，这个让我很郁闷，搞不清 `dev_open_window_fit...` 什么的函数是干嘛的？

x8515618:

因为你的用的是 VS2010，VS2010 使用的框架就是.NET Framework 4，所以你必须选择.NET Framework 4。假若你使用 vs2008，选择的框架就是.NET Framework 3.5，下面的以此类推了

reed1989:

谢谢指教，但是我没有搞清.NET Framework 4 和.NET Framework 4 Client Profile 是怎么回事，而且在新建工程的时候，就显示的是.NET Framework 4 没有后者，然后进去工程后，右击查看属性的时候却是.NET Framework 4 Client Profile。

x8515618:

http://www.cnblogs.com/St_Ding/archive/2011/11/23/2260050.html

这里讲的比较清楚了，你可以去看看

Visual Studio 2010 如期发布了，我怀着迫不及待的心情马上下载了最新的 ISO 来安装和感受一下。

.NET Framework 自从 2002 年发展至今，已经历了好几个版本，1.0, 1.1, 2.0, 3.0, 3.5 等不同的版本更替，.NET Framework 的 Redistributable Package 也愈来愈大，到了 .NET Framework 3.5 SP1 时已经膨胀到 231MB，除非软件本身就很大，不然对于 .NET 软件的 Redistribute 是相当不利的，除非客户端本身就有安装 .NET Framework，否则几乎都要夹带一个 .NET Framework 的 Redistributable Package，如果只是一个小小的应用程序文件（几十或几百 KB）还要夹一个那大的 .NET Framework Redistributable Package，大概很多开发人员都会觉得麻烦，就算开发人员接受，但在企业网络的 IT 管理人员就不一定会接受了 … 所以 .NET Framework 的减肥一直都是 .NET 开发人员的愿望与期待，而 .NET Framework Client Profile 就是微软针对这个需要的主要响应。既然 .NET 应用程序不一定会用到所有的 .NET Class Library，那微软就将 .NET Framework 部份不常用的功能由主函数库中抽离出来，让 Redistributable Package 可以缩小，有利于重新发布应用程序。

.NET Framework Client Profile 在 .NET 3.5 时首次和开发人员见面，但是碍于它只能利用 ClickOnce 部署方式安装，因此使用它的开发人员不多，且也有许多人反应希望将 .NET Framework Client Profile 能够利用光盘或其他储存媒体转散布，微软从善如流，将 .NET Framework 4.0 的 Client Profile 改成 Redistributable Package。

The .NET Framework 4 Client Profile 包含下列功能：

- Common Language Runtime (CLR)
- ClickOnce
- Windows Forms
- Windows Presentation Foundation (WPF)
- Windows Communication Foundation (WCF)
- Entity Framework
- Windows Workflow Foundation
- 语音
- XSLT 支援
- LINQ to SQL
- Entity Framework 与 WCF Data Services 执行期设计函式库
- Managed Extensibility Framework (MEF)
- 动态型别
- 并列程序设计功能，例如 Task Parallel Library (TPL), Parallel LINQ (PLINQ), and Coordination Data Structures (CDS)
- 除错客户端应用程序

下列功能不包含，要另外安装 .NET Framework 4.0 的完整版本：

- ASP.NET
- 进阶 Windows Communication Foundation (WCF) 功能
- .NET Framework Data Provider for Oracle
- 编译用 MSBuild
- Visual Studio 2010 的项目中，下列应用程序会默认使用 .NET Framework 4.0 Client Profile，若要使

用完整的 .NET Framework 功能，则要自行到项目属性中设定 target framework 至 .NET Framework 4.0：

- Windows Projects (C# and Visual Basic)
- Windows Presentation Foundation
- WPF Application
- WPF Browser Application
- WPF Custom Control Library
- WPF User Control Library
- Windows Forms
- Windows Forms Application
- Windows Forms Control Library
- Console Application
- Empty Project
- Window Service
- Visual F#
- F# Application
- F# Tutorial
- Workflow (C# and Visual Basic)
- Activity Designer Library
- Activity Library
- Workflow Console Application
- Windows Communication Foundation
- WCF Service Library
- Office 2007 and Office 2010
- All projects target the .NET Framework 4 Client Profile

而 .NET Framework 4.0 Client Profile 与原有的 .NET Framework 3.5 Client Profile 不同的地方有：

- 3.5 版本的 Client Profile 只能使用在 Web (ClickOnce)，4.0 版本则因为具有 Redistributable Package，所以可同时适用于本机或 Web 安装。
- 3.5 版本只支持 x86 的 Windows XP SP2/SP3，4.0 版本则支持除了 IA64 以外，.NET Framework 4 支持的所有平台。
- 3.5 版本的 Client Profile 与 .NET Framework 3.5 是分开的版本，4.0 版则为 .NET Framework 4.0 完整功能的子集。
- 3.5 版本会受 Windows Update 的影响自动升级到 .NET Framework 3.5，4.0 版则可以在不需要完整 .NET Framework 4.0 下自行运作，也可以分开提供服务。

不过也因为如此，当在 **Visual Studio 2010** 建立新项目时，如果发现 **[添加引用]** 中没有列出原本应该列出的组件或者引用的其他程序集，在代码中的调用一直提示找不到名字空间时，可以先检查是否是项目的 **target framework** 设成了 **.NET 4.0 Client Profile** (例如 **System.Web.dll** 就不在 **.NET 4.0 Client Profile** 套件中)，若是改成**.NET Framework 4** 就 **OK** 了。

huan_gong:

问一下：我的导出来的程序没有这个函数啊？“public void RunHalcon(HTuple Window)”
这个函数是你自己在 C# 代码中加的还是需要在 hdevelople 中写一行什么代码才能生成啊？

reed1989:

这个函数导出时就有呀。不过真正有用的函数是私有函数 **action()**，你可以把 **action()** 函数改为 **public**，然后自己把 **hv_ExpDefaultWinHandle** 赋为你的控件，直接调用 **action()** 也可以。

reed1989:

今天我遇到了引用外部函数程序了，我在使用 `disp_message()` 这个函数时，发现它是需要引用外部函数的，所以导出的时候必须将外部函数内容一起导出。导出的外部过程是 `public void disp_message (HTuple hv_WindowHandle, HTuple hv_String, HTuple hv_CoordSystem, HTuple hv_Row, HTuple hv_Column, HTuple hv_Color, HTuple hv_Box)` 这样一个函数，所以看来我以前那种随便勾选的导出代码是不对了。

灵峤 lencue:

是的，我一般都勾选前三项再导出的，这样各类函数都有了，直接就能用了。

huan_gong:

我的也导出来了，不大一样：

也是有一个类：`public partial class HDevelopExport`

里面有三个函数但分别是：

`public void dev_update_off()`

`public void set_display_font(HTuple hv_WindowHandle, HTuple hv_Size, HTuple hv_Font, HTuple hv_Bold, HTuple hv_Slant)` ----主要是设置窗体外观的

`private void action(HTuple hWindowControl1)`----主要代码都在这里

另外想问下：我的是通过一个跳出的窗口显示最终结果的，如何在和 `buttle` 同一个窗口显示啊？

这里都是把最终结果一次显示出来，如果需要显示中间过程怎么操作啊？（因为导出来就一个类，一下就从头到尾全部处理了，中间过程好像不好操作啊，请教下各位是怎么处理的啊？）

例如先匹配，然后测量，设置两个 `hWindowControl` 分别显示需要如何操作啊？

请教下：我在 `halcon` 写了如下代码：

```
dev_close_window()  
read_image (S1, 'D:/Koala.jpg')  
get_image_size(S1, Width, Height)  
dev_open_window (0, 0, Width/2, Height/2, 'black', WindowHandle)  
dev_display (S1)
```

导到 Csharp 后：会出现显示图片会跳出一个窗口来显示图片，而不是在 `form` 上的那 `hwindowcontrol` 显示；

把里面那句"`openwindow.....`"去掉后，就不会跳出来窗口显示，但是本身也不会显示；

把 `halcon` 代码改为如下：转过去到 C# 也不显示；

```
read_image (S1, 'D:/Koala.jpg')  
get_image_size(S1, Width, Height)  
dev_display (S1)
```

请问是怎么回事啊？

在 vs2010 界面中单击 帮助-管理帮助设置，然后弹出一个对话框



并没有选项供我选择安装在哪个目录，但是我在以前安装的时候却有，这次直接就这个对话框，让我很纳闷，请问，怎么才能在本地安装 helplibrary 呢？



在 csdn 论坛里找到了相关的解决方案，说找到文件"HelpLibManager.exe.config"，将"`<add key="FirstTimeRun" value="False" />`"中的 False 改为 True，我将这个值改成 True 后再安装，结果就弹出这个对话框了



请浏览到 helplibmanager.exe.config (C:\Program Files\Microsoft Help Viewer\v1.0\), 然后打开 helplibmanager.exe.config (在 notepad.exe 中), 再将 key="FirstTimeRun" value="False" 更改为 key="FirstTimeRun" value="True"。下次运行 Help Library 管理器时, 它将提醒您提供用于存储文档的文件夹路径。

安装完成 vs2010 后, 可以打开 Start\Programs\Microsoft Visual Studio 2010\Visual Studio Tools\Manage Help Settings 来安装。先选择安装位置, 然后选择是在线还是离线, 然后选择 Install from disk, 找到光盘中 \ProductDocumentation\HelpContentSetup.msha, 然后选择安装哪些项目的 library, 然后开始安装。

第一个 Blob 分析的例子，抽取 Blobs

看到群里有群友要求看个 Blob 分析的学习实例，正好这段时间开始 Blob 的测试，先发一个最基础的吧。

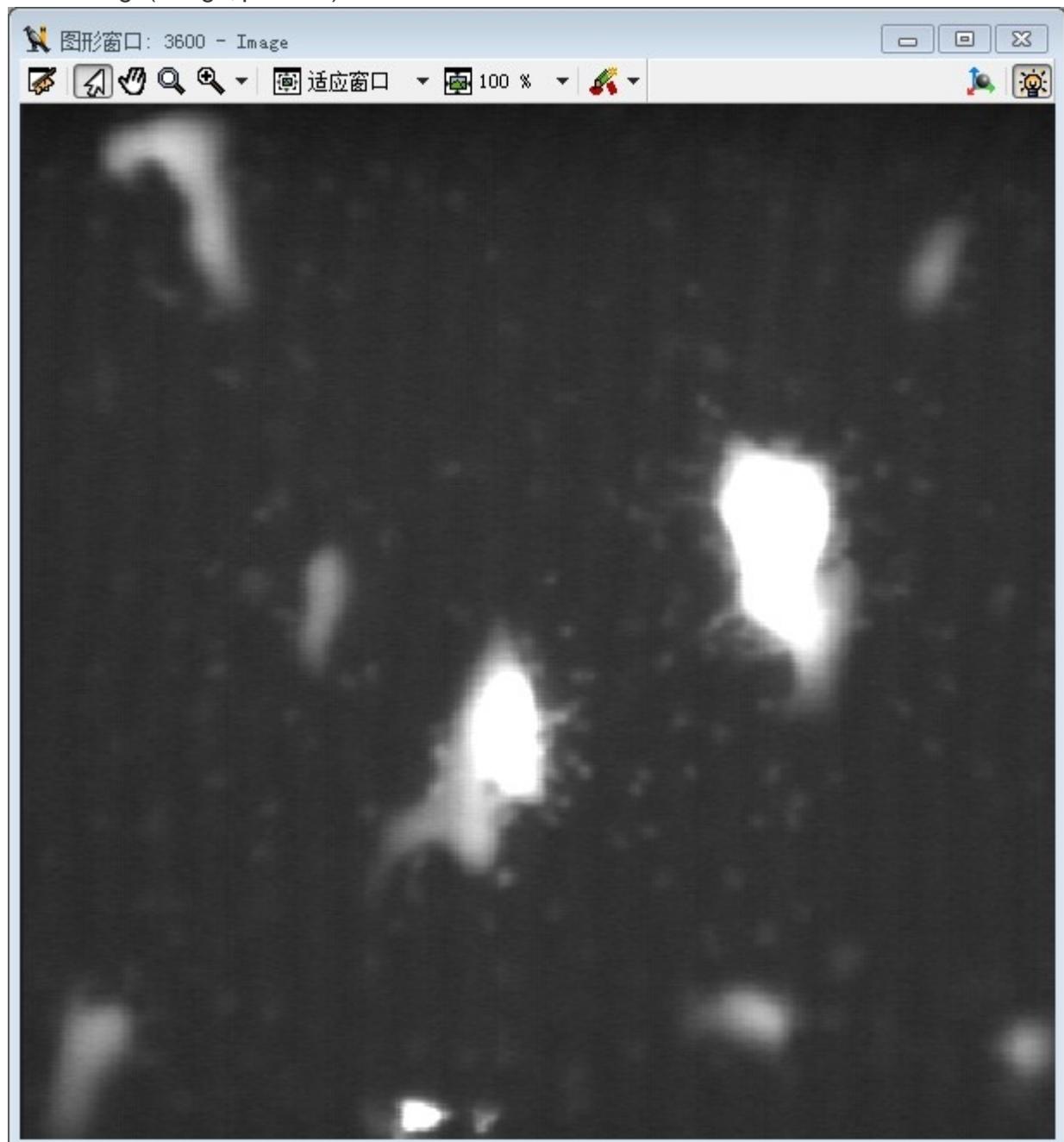
Extracted blobs with calculated center points (SG-i-P29)

Blob: Binary Large Object, 大型二进制对象。

在计算机中，一个大型二进制对象通常是一个大型的文件，例如一个图片或声音文件。而这些文件不得不以一种特殊的方式存放在数据库中。按照 Eric

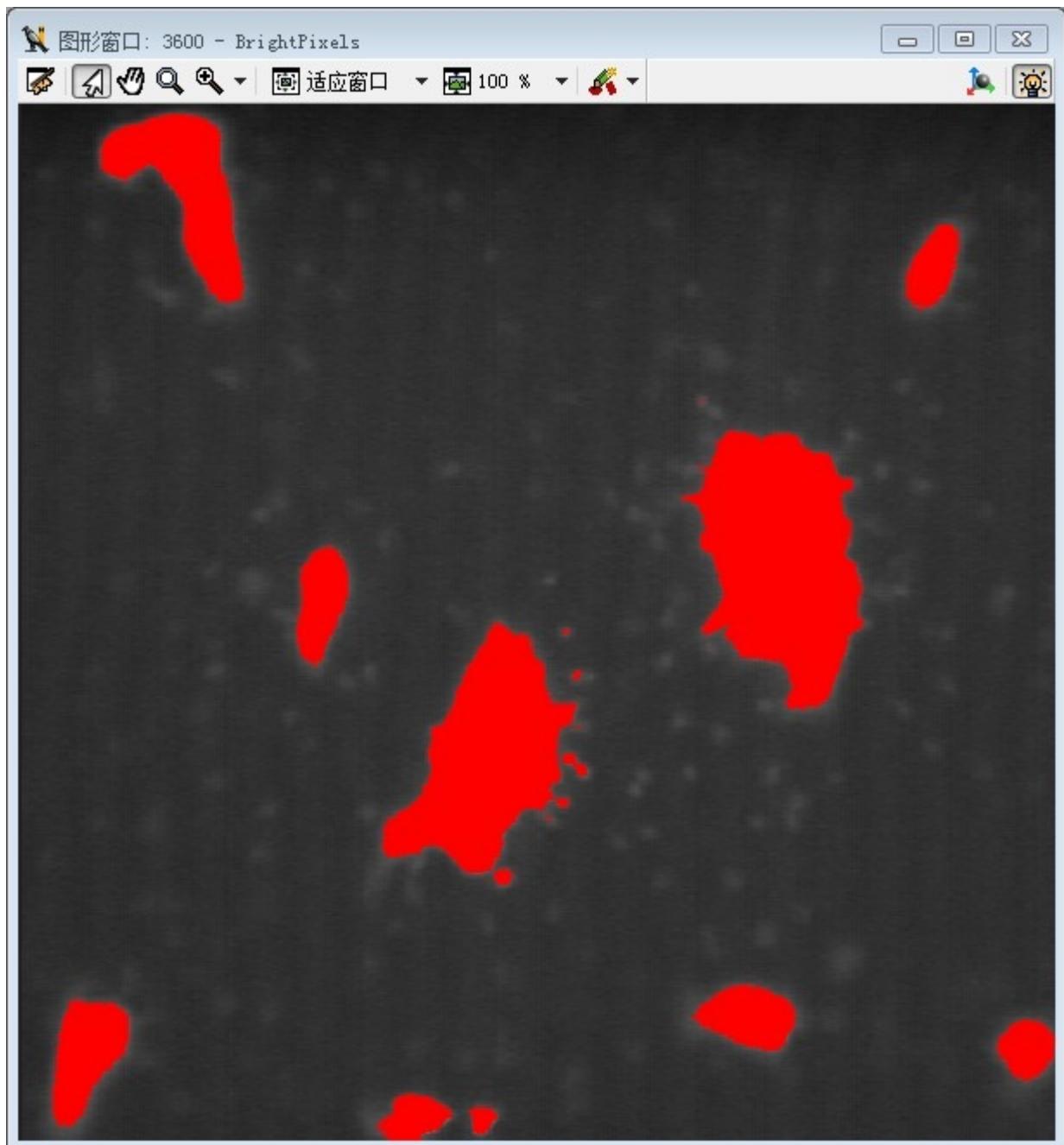
Raymond 的解释，大型二进制对象的意义在于，当文件的处理者（例如数据库管理工具）无法识别文件的格式以致无法处理该文件时，就把它作为大型二进制对象来处理。也有人说大型二进制对象实际上指代的是大型数据对象。一个需要处理大型二进制对象的应用程序是存储了大型多媒体对象（例如电影和电视节目）的数据存储器。

```
dev_close_window ()  
dev_open_window (0, 0, 600, 600, 'black',  
WindowHandle)  
read_image(Image,'particle')
```



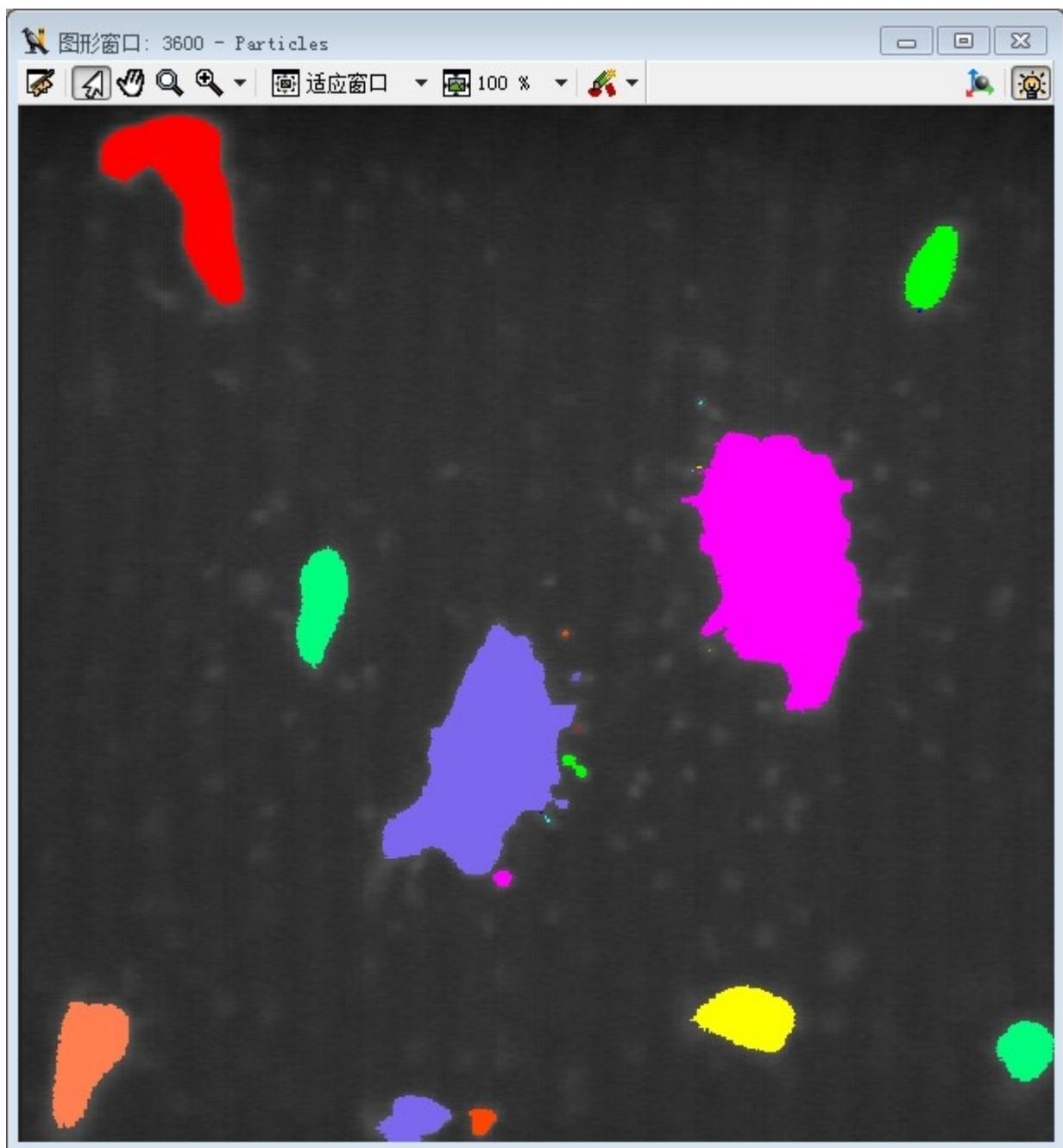
```
threshold(Image,BrightPixels,80,255)
```

- * 从获取的图片中选取灰度值为 80-255 的区域，这个区域命名为 BrightPixels；
- * 阈值分割出来的区域“Region”是根据灰度值范围分割的，那么只要是阈值范围内的区域都会被选取显示。



```
connection(BrightPixels,Particles)
```

- * 阈值分割出来的 Region，即这个例子中的 BrightPixels，把边缘相连的区域，即图中的单个红色阈值区域，
- * 连成一个区域，不相连的区域变成单独的区域，就相当于分成各个不同的区域块。



area_center(Particles, Area, Row, Column)

代码复制黏贴即可测试。

非常简单的一个初级实例，主要用到的是灰度值的取值范围。

Stage II's hdev 的代码实例

```
=====
* Create_read_find_matching_model_MK-skm.hdev
*
* 读取一个轮廓模型 (shape model), 使这个模型可放大或缩小地进行寻找匹配。
dev_close_window()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
read_image(Image,'L:/Halcon test/mk3.jpg')
* 选取一张有各种大小, 角度, 甚至变形的商标的图片做测试
*read_image(Image,'L:/Halcon test/mk5.jpg')
gen_rectangle1 (ROI, 57.8333, 49.5, 181.167, 342.833)
* 画一个矩形选择 ROI, 矩形在左上角, 覆盖一个完整的, 无变形规定尺寸的商标, 作为模板
reduce_domain(Image,ROI,ImageReduced)
* 大图和这个矩形的 ROI 相减就会得到一个左上角的商标的图案作为模板, 命名 ImageReduced
inspect_shape_model (ImageReduced, ModelImages, ModelRegions, 5, 40)
* 创建一个对比的模型, 后面两个参数: 第一个是层数 (NumLevels, 如 5 层), 第二个是
create_scaled_shape_model (ImageReduced, 5, rad(-180), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID)
* 创建一个比例不变的匹配的轮廓模型
* create_scaled_shape_model(Template : : NumLevels, AngleStart, AngleExtent, AngleStep, ScaleMin, ScaleMax,
ScaleStep, Optimization, Metric, Contrast, MinContrast : ModelID)
* 01、Template,//reduce_domain 后的模板图像
* 02、NumLevels,//金字塔的层数, 可设为“auto”或 0—10 的整数
* 03、AngleStart,//模板旋转的起始角度
* 04、AngleExtent,//模板旋转角度范围, >=0
* 05、AngleStep,//旋转角度的步长, >=0 and <=pi/16
* 06、ScaleMin,//模板最小比例
* 07、ScaleMax,//模板最大比例
* 08、ScaleStep,//模板比例的步长
* 09、Optimization,//设置模板优化和模板创建方法
* 10、Metric, //匹配方法设置
* 11、Contrast,/设置对比度
* 12、MinContrast,/设置最小对比度
* 13、ModelID,/输出模板句柄

* write_shape_model (ModelID, 'mk.shm')
* 把这个模型保存成一个 shm 模板
*clear_shape_model (ModelID)
stop()

*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image, ModelID, rad(-180), rad(360), 0.1, 5.0, 0.5, 0, 0.5, 'least_squares', 5, 0.8, Row,
Column, Angle, Scale, Score)
* 匹配一个比例不变的轮廓模型, 参数和创建的参数差不多
get_shape_model_contours (ModelContours, ModelID, 1)
for I := 0 to |Score|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row[I], Column[I], Angle[I], HomMat2DRotate)
    hom_mat2d_scale (HomMat2DRotate, Scale[I], Scale[I], Row[I], Column[I], HomMat2DScale)
```

```

affine_trans_contour_xld (ModelContours, ModelTrans, HomMat2DScale)
dev_display (ModelTrans)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score| + '个商标', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
* 但是变形则不能被识别
stop()
clear_shape_model (ModelID)

```

* Extracted blobs with calculated center points (SG-i-P29).hdev

```

*
dev_close_window ()
dev_open_window (0, 0, 300, 300, 'black', WindowHandle)
read_image(Image1,'particle')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
threshold(Image1,BrightPixels,80,255)
* 从获取的图片中选取灰度值为 80-255 的区域，这个区域命名为 BrightPixels;
* 阈值分割出来的区域“Region”是根据灰度值范围分割的，那么只要是阈值范围内的
* 区域都会被选取显示。
connection(BrightPixels,Particles)
* 阈值分割出来的 Region，即这个例子中的 BrightPixels，把边缘相连的区域，即图中的单个红色阈值区域，
* 连成一个区域，不相连的区域变成单独的区域，就相当于分成各个不同的区域块。
area_center(Particles, Area, Row, Column)
stop()

dev_clear_window()
read_image(Image2,'L:/Halcon test/mk2.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
threshold(Image2,BrightPixels,0,100)
connection(BrightPixels,Particles)
area_center(Particles, Area, Row, Column)
stop()

dev_clear_window()
read_image(Image3,'L:/Halcon test/mk3.jpg')
get_image_size (Image3, Width, Height)
dev_open_window_fit_image (Image3, 0, 0, Width, Height, WindowHandle)
dev_display(Image3)
gen_rectangle2 (ROI, 451.167, 384.5, rad(18.0042), 210.297, 110.003)
reduce_domain(Image3,ROI,ImageReduced)
threshold(ImageReduced,BrightPixels,0,10)
connection(BrightPixels,Particles)
* 读取，分色出这个 logo 中多少个不连接的区域
stop()

```

```

dev_clear_window()
read_image (Image4,'L:/Halcon test/pic.jpg')
get_image_size (Image4, Width, Height)
dev_open_window_fit_image (Image4, 0, 0, Width, Height, WindowHandle)
dev_display(Image4)
decompose3(Image4, Image11, Image22, Image33)
*转化成 RGB 三个通道图
rgb1_to_gray(Image4,GrayImage)
*转化成灰度图
threshold (Image4, Region, 128, 255)
*全局阈值

```

* Finding all clips in the image (MK).hdev

```

*
* 在一个图片中获取 ROI 并在此图片中匹配
dev_close_window ()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
*read_image(Image,'L:/Halcon test/mk2.jpg')
*read_image(Image,'L:/Halcon test/mk3.jpg')
*read_image(Image,'L:/Halcon test/mk4.jpg')
read_image(Image,'L:/Halcon test/mk5.jpg')
gen_rectangle1 (ROI1, 57.8333, 49.5, 181.167, 342.833)
reduce_domain(Image,ROI1,ImageReduced1)
create_shape_model(ImageReduced1,0,0,rad(360),0,'no_pregeneration','use_polarity',40,10,ModelID1)
find_shape_model(Image,ModelID1,0,rad(360),0.7,13,0.5,'interpolation',0,0.9,Row,Column,Angle,Score)
* 寻找与模板的大小尺寸必须是一比一匹配的，只是角度的不同而已，若大小发生变化，则不能匹配
get_shape_model_contours(ModelContours1,ModelID1,1)
for i := 0 to |Row|-1 by 1
    vector_angle_to_rigid(0,0,0,Row[i],Column[i],Angle[i],HomMat2D)
    affine_trans_contour_xld(ModelContours1,ContoursAffinTrans,HomMat2D)
endfor
disp_message (WindowHandle, '总共匹配了' + |Row| + '个商标', 'window', 12, 12, 'red', 'true')
clear_shape_model(ModelID1)
stop()

```

* 在一个图片中获取 ROI，并在另一个图片中匹配

```

dev_close_window ()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
read_image(Image1,'L:/Halcon test/mk1.jpg')
gen_rectangle1 (ROI2, 174.183, 54.3033, 295.347, 342.355)
reduce_domain(Image1,ROI2,ImageReduced2)
create_shape_model(ImageReduced2,0,0,rad(360),0,'no_pregeneration','use_polarity',40,10,ModelID2)
read_image(Image2,'L:/Halcon test/mk2.jpg')
find_shape_model(Image2,ModelID2,0,rad(360),0.7,13,0.5,'interpolation',0,0.9,Row,Column,Angle,Score)
* 寻找与模板的大小尺寸必须是一比一匹配的，只是角度的不同而已，若大小发生变化，则不能匹配
get_shape_model_contours(ModelContours2,ModelID2,1)
for i := 0 to |Row|-1 by 1

```

```
vector_angle_to_rigid(0,0,0,Row[i],Column[i],Angle[i],HomMat2D)
affine_trans_contour_xld(ModelContours2,ContoursAffinTrans,HomMat2D)
endfor
clear_shape_model(ModelID2)
```

* Finding all clips in the image (SG-i-P99).hdev

```
*
```

```
read_image(Image,'clip')
gen_rectangle2(ROI,124,181,0.653,129,47)
reduce_domain(Image,ROI,ImageReduced)
create_shape_model(ImageReduced,0,0,rad(360),0,'no_pregeneration','use_polarity',40,10,ModelID)
find_shape_model(Image,ModelID,0,rad(360),0.7,13,0.5,'interpolation',0,0.9,Row,Column,Angle,Score)
get_shape_model_contours(ModelContours,ModelID,1)
for i := 0 to |Row|-1 by 1
    vector_angle_to_rigid(0,0,0,Row[i],Column[i],Angle[i],HomMat2D)
    affine_trans_contour_xld(ModelContours,ContoursAffinTrans,HomMat2D)
endfor
clear_shape_model(ModelID)
```

* led and mianban matching.hdev

```
*
```

```
* 读取一个轮廓模型 (shape model), 使这个模型可放大或缩小地进行寻找匹配。
dev_close_window ()
* dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI, 41.5, 22.5, 104.5, 138.5)
reduce_domain(Image1,ROI,ImageReduced)
inspect_shape_model (ImageReduced, ModelImages, ModelRegions, 5, 40)
create_scaled_shape_model (ImageReduced, 5, rad(-180), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID)
* 创建一个比例可变的匹配的轮廓模型
* create_scaled_shape_model(Template : : NumLevels, AngleStart, AngleExtent, AngleStep, ScaleMin, ScaleMax,
ScaleStep, Optimization, Metric, Contrast, MinContrast : ModelID)
* 01、Template,//reduce_domain 后的模板图像
* 02、NumLevels,//金字塔的层数, 可设为“auto”或0—10的整数
* 03、AngleStart,//模板旋转的起始角度
* 04、AngleExtent,//模板旋转角度范围, >=0
* 05、AngleStep//旋转角度的步长, >=0 and <=pi/16
* 06、ScaleMin,//模板最小比例
* 07、ScaleMax,//模板最大比例
* 08、ScaleStep//模板比例的步长
* 09、Optimization,//设置模板优化和模板创建方法
* 10、Metric, //匹配方法设置
```

```

* 11、Contrast,//设置对比度
* 12、MinContrast,/设置最小对比度
* 13、ModelID,/输出模板句柄

* write_shape_model (ModelID, 'mk.shm')
* 把这个模型保存成一个 shm 模板
*clear_shape_model (ModelID)
stop()
dev_close_window ()
read_image(Image2,'L:/Halcon test/led/mianbanw.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID, rad(-180), rad(360), 0.1, 5.0, 0.5, 0, 0.5, 'least_squares', 5, 0.8, Row,
Column, Angle, Scale, Score)
* find_scaled_shape_model(Image, ModelID, AngleStart, AngleExtent, ScaleMin, ScaleMax, MinScore,
NumMatches, MaxOverlap, SubPixel, NumLevels, Greediness : Row, Column, Angle, Scale, Score)
* 匹配一个比例可变的轮廓模型，参数和创建的参数差不多
* Image :://搜索图像
* ModelID,/模板句柄
* AngleStart, // 搜索时的起始角度
* AngleExtent, //搜索时的角度范围，必须与创建模板时的有交集
* ScaleMin, //模板最小比例
* ScaleMax,/模板最大比例
* MinScore, //最小匹配值，输出的匹配的得分 Score 大于该值
* NumMatches, //定义要输出的匹配的最大个数
* MaxOverlap, //当找到的目标存在重叠时，且重叠大于该值时选择一个好的输出
* SubPixel, //计算精度的设置，五种模式，多选 2, 3
* NumLevels, //搜索时金字塔的层数
* Greediness : //贪婪度，搜索启发式，一般都设为 0.9，越高速度快,容易出现找不到的情况
* Row, Column, Angle, Score) //输出匹配位置的行和列坐标、角度、得分。

```

```

get_shape_model_contours (ModelContours, ModelID, 1)
for I := 0 to |Score|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row[I], Column[I], Angle[I], HomMat2DRotate)
    hom_mat2d_scale (HomMat2DRotate, Scale[I], Scale[I], Row[I], Column[I], HomMat2DScale)
    affine_trans_contour_xld (ModelContours, ModelTrans, HomMat2DScale)
    dev_display (ModelTrans)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score| + '个商标', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
* 但是变形则不能被识别
stop ()
clear_shape_model (ModelID)

```

```
=====
* multiple_models.hdev
*
* -----
* This example program searches for two types of objects simultaneously.
* -----
* general configuration of HDevelop
dev_update_window ('off')
* image acquisition and window size
read_image (ModelImage, 'rings_and_nuts')
get_image_pointer1 (ModelImage, Pointer, Type, Width, Height)
dev_close_window ()
dev_open_window (0, 0, Width, Height, 'white', WindowHandle)
dev_set_part (0, 0, Height-1, Width-1)
dev_display (ModelImage)
* colors and other settings for the visualization
dev_set_color ('blue')
dev_set_draw ('margin')
dev_set_line_width (2)
set_display_font (WindowHandle, 14, 'mono', 'true', 'false')
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
* ----- start of the application -----
* -> select the model objects
gen_circle (ModelROIRing, 121, 299, 55)
gen_circle (ModelROINut, 324, 279, 52)
dev_display (ModelImage)
dev_display (ModelROIRing)
dev_display (ModelROINut)
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
* step 1: create the models
reduce_domain (ModelImage, ModelROIRing, ImageROIRing)
create_scaled_shape_model (ImageROIRing, 'auto', -rad(22.5), rad(45), 'auto', 0.8, 1.2, 'auto', 'none', 'use_polarity', 60, 10, ModelIDRing)
inspect_shape_model (ImageROIRing, PyramidImage, ModelRegionRing, 1, 30)
get_shape_model_contours (ShapeModelRing, ModelIDRing, 1)
count_obj (ShapeModelRing, NumContoursRing)
dev_clear_window ()
dev_display (ModelRegionRing)
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
reduce_domain (ModelImage, ModelROINut, ImageROINut)
create_scaled_shape_model (ImageROINut, 'auto', -rad(30), rad(60), 'auto', 0.6, 1.4, 'auto', 'none', 'use_polarity', 60, 10, ModelIDNut)
inspect_shape_model (ImageROINut, PyramidImage, ModelRegionNut, 1, 30)
get_shape_model_contours (ShapeModelNut, ModelIDNut, 1)
count_obj (ShapeModelNut, NumContoursNut)
dev_display (ModelRegionNut)
```

```

* step 2: save the relevant information in tuples
ModelIDs := [ModelIDRing, ModelIDNut]
concat_obj (ShapeModelRing, ShapeModelNut, ShapeModels)
StartContoursInTuple := [1, NumContoursRing+1]
NumContoursInTuple := [NumContoursRing, NumContoursNut]
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
read_image (SearchImage, 'rings_and_nuts')
dev_display (SearchImage)

* step 3: create two search ROIs (upper and lower part of the image)
gen_rectangle1 (SearchROIRing, 110, 10, 130, Width - 10)
gen_rectangle1 (SearchROIINut, 315, 10, 335, Width - 10)
* -> "add" the ROIs to the search image
concat_obj (SearchROIRing, SearchROIINut, SearchROIs)
dev_set_color ('blue')
dev_display (SearchROIs)
add_channels (SearchROIs, SearchImage, SearchImageReduced)
disp_continue_message (WindowHandle, 'black', 'true')
stop ()

* step 4: find all objects
dev_display (SearchImage)
dev_set_color ('cyan')
find_scaled_shape_models (SearchImageReduced, ModelIDs, [-rad(22.5), -rad(30)], [rad(45), rad(60)], [0.8, 0.6], [1.2, 1.4], 0.7, 0, 0, 'least_squares', 0, 0.8, RowCheck, ColumnCheck, AngleCheck, ScaleCheck, Score, ModelIndex)
for i := 0 to |Score| - 1 by 1
    * step 4: determine the transformation
    Model := ModelIndex[i]
    vector_angle_to_rigid (0, 0, 0, RowCheck[i], ColumnCheck[i], AngleCheck[i], MovementOfObject)
    hom_mat2d_scale (MovementOfObject, ScaleCheck[i], ScaleCheck[i], RowCheck[i], ColumnCheck[i], MoveAndScalingOfObject)
    * -> access the XLD contours belonging to the model
    copy_obj (ShapeModels, ShapeModel, StartContoursInTuple[Model], NumContoursInTuple[Model])
    affine_trans_contour_xld (ShapeModel, ModelAtNewPosition, MoveAndScalingOfObject)
    dev_display (ModelAtNewPosition)
    disp_continue_message (WindowHandle, 'black', 'true')
    stop ()
endfor
* ----- end of the application -----
* clean up
dev_update_window ('on')
clear_shape_model (ModelIDRing)
clear_shape_model (ModelIDNut)

=====

* Processing the image only within the circular ROI (SG-i-P20).hdev
*
dev_close_window ()
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)

```

```

read_image(Image,'mreut')
gen_circle(ROI,256,256,200)
* 在行 256, 列 256 为圆心的位置, 画一个直径为 200 的园
reduce_domain(Image,ROI,ImageReduced)
* 在整个图片 (Image) 抠(图)出一个感兴趣区域 (ROI, 就刚才画的园), 命名为 ImageReduced
edges_sub_pix(ImageReduced,Edges,'lanser2',0.5,20,40)
dev_display(Image)
dev_clear_window()
dev_display(ROI)
dev_display(ImageReduced)
dev_display(Edges)

```

* led and mianban matching_n.hdev

```

* 读取一个轮廓模型 (shape model), 使这个模型可放大或缩小地进行寻找匹配。
* ~~~~~开始读取模板~~~~~
dev_close_window ()
* dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
* =====缺水, qs=====
gen_rectangle1 (ROI_qs, 41.5, 22.5, 104.5, 138.5)
* gen_rectangle1 (ROI, 117.5, 21.5, 189.5, 187.5)
reduce_domain(Image1,ROI_qs,ImageReduced_qs)
inspect_shape_model (ImageReduced_qs, ModelImages_qs, ModelRegions_qs, 5, 40)
create_scaled_shape_model (ImageReduced_qs, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_qs)
*write_shape_model (ModelID_qs, 'qs.shm')
* 把这个模型保存成一个 shm 模板
*clear_shape_model (ModelID_qs)
*stop()
* =====低水位, dsw=====
dev_close_window ()
*read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI_dsw, 117.5, 21.5, 189.5, 187.5)
reduce_domain(Image1,ROI_dsw,ImageReduced_dsw)
inspect_shape_model (ImageReduced_dsw, ModelImages_dsw, ModelRegions_dsw, 5, 40)
create_scaled_shape_model (ImageReduced_dsw, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_dsw)
*write_shape_model (ModelID_dsw, 'dsw.shm')
*clear_shape_model (ModelID_dsw)
* stop()
* =====定时关机, dsgj=====

```

```
dev_close_window ()  
*read_image(Image1,'L:/Halcon test/led/ledw.jpg')  
get_image_size (Image1, Width, Height)  
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)  
dev_display(Image1)  
gen_rectangle1 (ROI_dsgj, 219.5, 23.5, 294.5, 239.5)  
reduce_domain(Image1,ROI_dsgj,ImageReduced_dsgj)  
inspect_shape_model (ImageReduced_dsgj, ModelImages_dsgj, ModelRegions_dsgj, 5, 40)  
create_scaled_shape_model (ImageReduced_dsgj, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],  
'ignore_global_polarity', 40, 10, ModelID_dsgj)  
*write_shape_model (ModelID_dsgj, 'dsgj.shm')  
*clear_shape_model (ModelID_dsgj)  
*stop()  
* =====30°C 睡眠, sleep=====  
dev_close_window ()  
* read_image(Image1,'L:/Halcon test/led/ledw.jpg')  
get_image_size (Image1, Width, Height)  
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)  
dev_display(Image1)  
gen_rectangle1 (ROI_sleep, 26.5, 193.5, 200.5, 488.5)  
reduce_domain(Image1,ROI_sleep,ImageReduced_sleep)  
inspect_shape_model (ImageReduced_sleep, ModelImages_sleep, ModelRegions_sleep, 5, 40)  
create_scaled_shape_model (ImageReduced_sleep, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],  
'ignore_global_polarity', 40, 10, ModelID_sleep)  
*write_shape_model (ModelID_sleep, 'sleep.shm')  
*clear_shape_model (ModelID_sleep)  
*stop()  
* =====定时关机时间, time=====  
dev_close_window ()  
* read_image(Image1,'L:/Halcon test/led/ledw.jpg')  
get_image_size (Image1, Width, Height)  
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)  
dev_display(Image1)  
gen_rectangle1 (ROI_time, 208.5, 253.5, 301.5, 486.5)  
reduce_domain(Image1,ROI_time,ImageReduced_time)  
inspect_shape_model (ImageReduced_time, ModelImages_time, ModelRegions_time, 5, 40)  
create_scaled_shape_model (ImageReduced_time, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],  
'ignore_global_polarity', 40, 10, ModelID_time)  
*write_shape_model (ModelID_time, 'time.shm')  
*clear_shape_model (ModelID_time)  
*stop()  
* =====定时关机+时间, dsgjtime=====  
dev_close_window ()  
* read_image(Image1,'L:/Halcon test/led/ledw.jpg')  
get_image_size (Image1, Width, Height)  
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)  
dev_display(Image1)  
gen_rectangle1 (ROI_dsgjtime, 208.5, 21.5, 301.5, 486.5)  
reduce_domain(Image1,ROI_dsgjtime,ImageReduced_dsgjtime)
```

```
inspect_shape_model (ImageReduced_dsgjtime, ModelImages_dsgjtime, ModelRegions_dsgjtime, 5, 40)
create_scaled_shape_model (ImageReduced_dsgjtime, 5, rad(0), rad(360), 0, 0.1, 5.0, 0,
['none','no_pregeneration'], 'ignore_global_polarity', 40, 10, ModelID_dsgjtime)
*write_shape_model (ModelID_dsgjtime, 'dsgjtime.shm')
*clear_shape_model (ModelID_dsgjtime)
stop()
* ~~~~~结束读取模板~~~~~
```

```
* ~~~~~开始寻找匹配~~~~~
```

```
dev_close_window ()
read_image(Image2,'L:/Halcon test/led/mianban.jpg')
*read_image(Image2,'L:/Halcon test/led/mianbanw.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
```

```
* =====缺水, qs=====
```

```
find_scaled_shape_model (Image2, ModelID_qs, rad(0), rad(360), 0.1, 5.0, 0.5, 0, 0.5, 'least_squares', 5, 0.8,
Row_qs, Column_qs, Angle_qs, Scale_qs, Score_qs)
```

```
* find_scaled_shape_model(Image, ModelID, AngleStart, AngleExtent, ScaleMin, ScaleMax, MinScore,
NumMatches, MaxOverlap, SubPixel, NumLevels, Greediness : Row, Column, Angle, Scale, Score)
```

```
* 匹配一个比例可变的轮廓模型, 参数和创建的参数差不多
```

```
* Image :: //搜索图像
```

```
* ModelID, //模板句柄
```

```
* AngleStart, // 搜索时的起始角度
```

```
* AngleExtent, //搜索时的角度范围, 必须与创建模板时的有交集
```

```
* ScaleMin, //模板最小比例
```

```
* ScaleMax, //模板最大比例
```

```
* MinScore, //最小匹配值, 输出的匹配的得分 Score 大于该值
```

```
* NumMatches, //定义要输出的匹配的最大个数
```

```
* MaxOverlap, //当找到的目标存在重叠时, 且重叠大于该值时选择一个好的输出
```

```
* SubPixel, //计算精度的设置, 五种模式, 多选 2, 3
```

```
* NumLevels, //搜索时金字塔的层数
```

```
* Greediness : //贪婪度, 搜索启发式, 一般都设为 0.9, 越高速度快, 容易出现找不到的情况
```

```
* Row, Column, Angle, Score) //输出匹配位置的行和列坐标、角度、得分。
```

```
get_shape_model_contours (ModelContours_qs, ModelID_qs, 1)
```

```
for I := 0 to |Score_qs|-1 by 1
```

```
    vector_angle_to_rigid (0, 0, 0, Row_qs[I], Column_qs[I], Angle_qs[I], HomMat2DRotate_qs)
```

```
    hom_mat2d_scale (HomMat2DRotate_qs, Scale_qs[I], Scale_qs[I], Row_qs[I], Column_qs[I],
HomMat2DScale_qs)
```

```
    affine_trans_contour_xld (ModelContours_qs, ModelTrans_qs, HomMat2DScale_qs)
```

```
    dev_display (ModelTrans_qs)
```

```
endfor
```

```
disp_message (WindowHandle, '总共匹配了' + |Score_qs| + '个符号', 'window', 12, 12, 'red', 'true')
```

```
disp_continue_message (WindowHandle, 'black', 'true')
```

```
* 但是变形则不能被识别
```

```

*stop()
wait_seconds(2)

* =====低水位, dsw=====
dev_close_window()
read_image(Image2,'L:/Halcon test/led/led2.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_dsw, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,3], 0.8,
Row_dsw, Column_dsw, Angle_dsw, Scale_dsw, Score_dsw)
get_shape_model_contours (ModelContours_dsw, ModelID_dsw, 1)
for I := 0 to |Score_dsw|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_dsw[I], Column_dsw[I], Angle_dsw[I], HomMat2DRotate_dsw)
    hom_mat2d_scale (HomMat2DRotate_dsw, Scale_dsw[I], Scale_dsw[I], Row_dsw[I], Column_dsw[I],
HomMat2DScale_dsw)
    affine_trans_contour_xld (ModelContours_dsw, ModelTrans_dsw, HomMat2DScale_dsw)
    dev_display (ModelTrans_dsw)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_dsw| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop()
wait_seconds(2)

```

```

* =====定时关机, dsgj=====
dev_close_window()
read_image(Image2,'L:/Halcon test/led/led2.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_dsgj, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8,
Row_dsgj, Column_dsgj, Angle_dsgj, Scale_dsgj, Score_dsgj)
get_shape_model_contours (ModelContours_dsgj, ModelID_dsgj, 1)
for I := 0 to |Score_dsgj|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_dsgj[I], Column_dsgj[I], Angle_dsgj[I], HomMat2DRotate_dsgj)
    hom_mat2d_scale (HomMat2DRotate_dsgj, Scale_dsgj[I], Scale_dsgj[I], Row_dsgj[I], Column_dsgj[I],
HomMat2DScale_dsgj)
    affine_trans_contour_xld (ModelContours_dsgj, ModelTrans_dsgj, HomMat2DScale_dsgj)
    dev_display (ModelTrans_dsgj)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_dsgj| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop()
wait_seconds(2)

```

* =====30°C 睡眠, sleep=====

```

dev_close_window()
read_image(Image2,'L:/Halcon test/led/bianm500.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_sleep, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8, Row_sleep, Column_sleep, Angle_sleep, Scale_sleep, Score_sleep)
get_shape_model_contours (ModelContours_sleep, ModelID_sleep, 1)
for I := 0 to |Score_sleep|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_sleep[I], Column_sleep[I], Angle_sleep[I], HomMat2DRotate_sleep)
    hom_mat2d_scale (HomMat2DRotate_sleep, Scale_sleep[I], Scale_sleep[I], Row_sleep[I], Column_sleep[I], HomMat2DScale_sleep)
    affine_trans_contour_xld (ModelContours_sleep, ModelTrans_sleep, HomMat2DScale_sleep)
    dev_display (ModelTrans_sleep)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_sleep| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop ()
wait_seconds(2)

```

```

* =====定时关机时间, time=====
dev_close_window()
read_image(Image2,'L:/Halcon test/led/bianm500.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_time, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8, Row_time, Column_time, Angle_time, Scale_time, Score_time)
get_shape_model_contours (ModelContours_time, ModelID_time, 1)
for I := 0 to |Score_time|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_time[I], Column_time[I], Angle_time[I], HomMat2DRotate_time)
    hom_mat2d_scale (HomMat2DRotate_time, Scale_time[I], Scale_time[I], Row_time[I], Column_time[I], HomMat2DScale_time)
    affine_trans_contour_xld (ModelContours_time, ModelTrans_time, HomMat2DScale_time)
    dev_display (ModelTrans_time)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_time| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop ()
wait_seconds(2)

```

```

* =====定时关机+时间, dsgjtime=====
dev_close_window()
read_image(Image2,'L:/Halcon test/led/bianm500.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)

```

```

*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_dsgjtime, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4],
0.8, Row_dsgjtime, Column_dsgjtime, Angle_dsgjtime, Scale_dsgjtime, Score_dsgjtime)
get_shape_model_contours (ModelContours_dsgjtime, ModelID_dsgjtime, 1)
for I := 0 to |Score_dsgjtime|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_dsgjtime[I], Column_dsgjtime[I], Angle_dsgjtime[I],
HomMat2DRotate_dsgjtime)
    hom_mat2d_scale (HomMat2DRotate_dsgjtime, Scale_dsgjtime[I], Scale_dsgjtime[I], Row_dsgjtime[I],
Column_dsgjtime[I], HomMat2DScale_dsgjtime)
    affine_trans_contour_xld (ModelContours_dsgjtime, ModelTrans_dsgjtime, HomMat2DScale_dsgjtime)
    dev_display (ModelTrans_dsgjtime)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_dsgjtime| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
stop ()

```

* ======30°C睡眠, sleep 错误匹配测试=====

```

dev_close_window ()
read_image(Image2,'L:/Halcon test/led/bianm500wrong.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_sleep, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4],
0.8, Row_sleep, Column_sleep, Angle_sleep, Scale_sleep, Score_sleep)
get_shape_model_contours (ModelContours_sleep, ModelID_sleep, 1)
for I := 0 to |Score_sleep|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_sleep[I], Column_sleep[I], Angle_sleep[I], HomMat2DRotate_sleep)
    hom_mat2d_scale (HomMat2DRotate_sleep, Scale_sleep[I], Scale_sleep[I], Row_sleep[I], Column_sleep[I],
HomMat2DScale_sleep)
    affine_trans_contour_xld (ModelContours_sleep, ModelTrans_sleep, HomMat2DScale_sleep)
    dev_display (ModelTrans_sleep)
endfor

```

disp_message (WindowHandle, '总共匹配了' + |Score_sleep| + '个符号', 'window', 12, 12, 'red', 'true')

disp_continue_message (WindowHandle, 'black', 'true')

*stop ()

wait_seconds(2)

* ======定时关机时间, time 错误匹配测试=====

```

dev_close_window ()
read_image(Image2,'L:/Halcon test/led/bianm500wrong.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
*read_shape_model ('mk.shm', ModelID)
find_scaled_shape_model (Image2, ModelID_time, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8,
Row_time, Column_time, Angle_time, Scale_time, Score_time)
get_shape_model_contours (ModelContours_time, ModelID_time, 1)
for I := 0 to |Score_time|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_time[I], Column_time[I], Angle_time[I], HomMat2DRotate_time)

```

```

hom_mat2d_scale (HomMat2DRotate_time, Scale_time[I], Scale_time[I], Row_time[I], Column_time[I],
HomMat2DScale_time)
affine_trans_contour_xld (ModelContours_time, ModelTrans_time, HomMat2DScale_time)
dev_display (ModelTrans_time)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_time| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
stop ()
*=====错误匹配测试失败，此匹配方案不能鉴别断笔缺符号的错误!!! =====
* ~~~~~结束寻找匹配~~~~
clear_shape_model (ModelID_dsw)
clear_shape_model (ModelID_qs)
clear_shape_model (ModelID_dsgj)
clear_shape_model (ModelID_dsgjtime)
clear_shape_model (ModelID_time)
clear_shape_model (ModelID_sleep)

=====

```

* KN20-LED-Matching-A: write_shape_model.hdev

```

* ~~~~~开始读取模板~~~~
dev_close_window ()
* dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
* =====缺水，qs=====
gen_rectangle1 (ROI_qs, 41.5, 22.5, 104.5, 138.5)
* gen_rectangle1 (ROI, 117.5, 21.5, 189.5, 187.5)
reduce_domain(Image1,ROI_qs,ImageReduced_qs)
inspect_shape_model (ImageReduced_qs, ModelImages_qs, ModelRegions_qs, 5, 40)
create_scaled_shape_model (ImageReduced_qs, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_qs)
write_shape_model (ModelID_qs, 'qs.shm')
* 把这个模型保存成一个 shm 模板
clear_shape_model (ModelID_qs)
*stop()
* =====低水位，dsw=====
dev_close_window ()
*read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI_dsw, 117.5, 21.5, 189.5, 187.5)
reduce_domain(Image1,ROI_dsw,ImageReduced_dsw)
inspect_shape_model (ImageReduced_dsw, ModelImages_dsw, ModelRegions_dsw, 5, 40)
create_scaled_shape_model (ImageReduced_dsw, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_dsw)

```

```
write_shape_model (ModelID_dsw, 'dsw.shm')
clear_shape_model (ModelID_dsw)
* stop()
* =====定时关机, dsgj=====
dev_close_window ()
*read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI_dsgj, 219.5, 23.5, 294.5, 239.5)
reduce_domain(Image1,ROI_dsgj,ImageReduced_dsgj)
inspect_shape_model (ImageReduced_dsgj, ModelImages_dsgj, ModelRegions_dsgj, 5, 40)
create_scaled_shape_model (ImageReduced_dsgj, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_dsgj)
write_shape_model (ModelID_dsgj, 'dsgj.shm')
clear_shape_model (ModelID_dsgj)
*stop()
* =====30°C睡眠, sleep=====
dev_close_window ()
* read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI_sleep, 26.5, 193.5, 200.5, 488.5)
reduce_domain(Image1,ROI_sleep,ImageReduced_sleep)
inspect_shape_model (ImageReduced_sleep, ModelImages_sleep, ModelRegions_sleep, 5, 40)
create_scaled_shape_model (ImageReduced_sleep, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_sleep)
write_shape_model (ModelID_sleep, 'sleep.shm')
clear_shape_model (ModelID_sleep)
*stop()
* =====定时关机时间, time=====
dev_close_window ()
* read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI_time, 208.5, 253.5, 301.5, 486.5)
reduce_domain(Image1,ROI_time,ImageReduced_time)
inspect_shape_model (ImageReduced_time, ModelImages_time, ModelRegions_time, 5, 40)
create_scaled_shape_model (ImageReduced_time, 5, rad(0), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'],
'ignore_global_polarity', 40, 10, ModelID_time)
write_shape_model (ModelID_time, 'time.shm')
clear_shape_model (ModelID_time)
*stop()
* =====定时关机+时间, dsgjtime=====
dev_close_window ()
* read_image(Image1,'L:/Halcon test/led/ledw.jpg')
get_image_size (Image1, Width, Height)
```

```

dev_open_window_fit_image (Image1, 0, 0, Width, Height, WindowHandle)
dev_display(Image1)
gen_rectangle1 (ROI_dsgjtime, 208.5, 21.5, 301.5, 486.5)
reduce_domain(Image1, ROI_dsgjtime, ImageReduced_dsgjtime)
inspect_shape_model (ImageReduced_dsgjtime, ModelImages_dsgjtime, ModelRegions_dsgjtime, 5, 40)
create_scaled_shape_model (ImageReduced_dsgjtime, 5, rad(0), rad(360), 0, 0.1, 5.0, 0,
['none', 'no_pregeneration'], 'ignore_global_polarity', 40, 10, ModelID_dsgjtime)
write_shape_model (ModelID_dsgjtime, 'dsgjtime.shm')
clear_shape_model (ModelID_dsgjtime)
stop()
* ~~~~~结束读取模板~~~~~
```

* KN20-LED-Matching-A: find_from_shape_model.hdev

```

* ~~~~~开始寻找匹配~~~~~
dev_close_window ()
read_image(Image2, 'L:/Halcon test/led/mianban.jpg')
*read_image(Image2, 'L:/Halcon test/led/mianbanw.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
read_shape_model ('qs.shm', ModelID_qs)

* =====缺水, qs=====
find_scaled_shape_model (Image2, ModelID_qs, rad(0), rad(360), 0.1, 5.0, 0.5, 0, 0.5, 'least_squares', 5, 0.8,
Row_qs, Column_qs, Angle_qs, Scale_qs, Score_qs)
* find_scaled_shape_model (Image, ModelID, AngleStart, AngleExtent, ScaleMin, ScaleMax, MinScore,
NumMatches, MaxOverlap, SubPixel, NumLevels, Greediness : Row, Column, Angle, Scale, Score)
* 匹配一个比例可变的轮廓模型, 参数和创建的参数差不多
* Image :: //搜索图像
* ModelID, //模板句柄
* AngleStart, // 搜索时的起始角度
* AngleExtent, //搜索时的角度范围, 必须与创建模板时的有交集
* ScaleMin, //模板最小比例
* ScaleMax, //模板最大比例
* MinScore, //最小匹配值, 输出的匹配的得分 Score 大于该值
* NumMatches, //定义要输出的匹配的最大个数
* MaxOverlap, //当找到的目标存在重叠时, 且重叠大于该值时选择一个好的输出
* SubPixel, //计算精度的设置, 五种模式, 多选 2, 3
* NumLevels, //搜索时金字塔的层数
* Greediness : //贪婪度, 搜索启发式, 一般都设为 0.9, 越高速度快, 容易出现找不到的情况
* Row, Column, Angle, Score) //输出匹配位置的行和列坐标、角度、得分。
```

```

get_shape_model_contours (ModelContours_qs, ModelID_qs, 1)
for I := 0 to |Score_qs|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_qs[I], Column_qs[I], Angle_qs[I], HomMat2DRotate_qs)
    hom_mat2d_scale (HomMat2DRotate_qs, Scale_qs[I], Scale_qs[I], Row_qs[I], Column_qs[I],
HomMat2DScale_qs)
```

```

affine_trans_contour_xld (ModelContours_qs, ModelTrans_qs, HomMat2DScale_qs)
dev_display (ModelTrans_qs)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_qs| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
* 但是变形则不能被识别
*stop ()
wait_seconds(2)

* =====低水位, dsw=====
dev_close_window ()
read_image(Image2,'L:/Halcon test/led/led2.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
read_shape_model ('dsw.shm', ModelID_dsw)
find_scaled_shape_model (Image2, ModelID_dsw, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,3], 0.8,
Row_dsw, Column_dsw, Angle_dsw, Scale_dsw, Score_dsw)
get_shape_model_contours (ModelContours_dsw, ModelID_dsw, 1)
for I := 0 to |Score_dsw|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_dsw[I], Column_dsw[I], Angle_dsw[I], HomMat2DRotate_dsw)
    hom_mat2d_scale (HomMat2DRotate_dsw, Scale_dsw[I], Scale_dsw[I], Row_dsw[I], Column_dsw[I],
HomMat2DScale_dsw)
    affine_trans_contour_xld (ModelContours_dsw, ModelTrans_dsw, HomMat2DScale_dsw)
    dev_display (ModelTrans_dsw)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_dsw| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop ()
wait_seconds(2)

* =====定时关机, dsgj=====
dev_close_window ()
read_image(Image2,'L:/Halcon test/led/led2.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
read_shape_model ('dsgj.shm', ModelID_dsgj)
find_scaled_shape_model (Image2, ModelID_dsgj, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8,
Row_dsgj, Column_dsgj, Angle_dsgj, Scale_dsgj, Score_dsgj)
get_shape_model_contours (ModelContours_dsgj, ModelID_dsgj, 1)
for I := 0 to |Score_dsgj|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_dsgj[I], Column_dsgj[I], Angle_dsgj[I], HomMat2DRotate_dsgj)
    hom_mat2d_scale (HomMat2DRotate_dsgj, Scale_dsgj[I], Scale_dsgj[I], Row_dsgj[I], Column_dsgj[I],
HomMat2DScale_dsgj)
    affine_trans_contour_xld (ModelContours_dsgj, ModelTrans_dsgj, HomMat2DScale_dsgj)
    dev_display (ModelTrans_dsgj)
endfor

```

```

disp_message (WindowHandle, '总共匹配了' + |Score_dsgj| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop ()
wait_seconds(2)

* ======30℃睡眠, sleep=====
dev_close_window ()
read_image(Image2,'L:/Halcon test/led/bianm500.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
read_shape_model ('sleep.shm', ModelID_sleep)
find_scaled_shape_model (Image2, ModelID_sleep, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8, Row_sleep, Column_sleep, Angle_sleep, Scale_sleep, Score_sleep)
get_shape_model_contours (ModelContours_sleep, ModelID_sleep, 1)
for I := 0 to |Score_sleep|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_sleep[I], Column_sleep[I], Angle_sleep[I], HomMat2DRotate_sleep)
    hom_mat2d_scale (HomMat2DRotate_sleep, Scale_sleep[I], Scale_sleep[I], Row_sleep[I], Column_sleep[I], HomMat2DScale_sleep)
    affine_trans_contour_xld (ModelContours_sleep, ModelTrans_sleep, HomMat2DScale_sleep)
    dev_display (ModelTrans_sleep)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_sleep| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop ()
wait_seconds(2)

* ======定时关机时间, time=====
dev_close_window ()
read_image(Image2,'L:/Halcon test/led/bianm500.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
read_shape_model ('time.shm', ModelID_time)
find_scaled_shape_model (Image2, ModelID_time, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8, Row_time, Column_time, Angle_time, Scale_time, Score_time)
get_shape_model_contours (ModelContours_time, ModelID_time, 1)
for I := 0 to |Score_time|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_time[I], Column_time[I], Angle_time[I], HomMat2DRotate_time)
    hom_mat2d_scale (HomMat2DRotate_time, Scale_time[I], Scale_time[I], Row_time[I], Column_time[I], HomMat2DScale_time)
    affine_trans_contour_xld (ModelContours_time, ModelTrans_time, HomMat2DScale_time)
    dev_display (ModelTrans_time)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_time| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
*stop ()
wait_seconds(2)

```

```

* =====定时关机+时间, dsgjtime=====
dev_close_window()
read_image(Image2,'L:/Halcon test/led/bianm500.jpg')
get_image_size (Image2, Width, Height)
dev_open_window_fit_image (Image2, 0, 0, Width, Height, WindowHandle)
dev_display(Image2)
read_shape_model ('dsgjtime.shm', ModelID_dsgjtime)
find_scaled_shape_model (Image2, ModelID_dsgjtime, rad(0), rad(360), 0.1, 5.0, 0.5, 1, 0.5, 'least_squares', [5,4], 0.8, Row_dsgjtime, Column_dsgjtime, Angle_dsgjtime, Scale_dsgjtime, Score_dsgjtime)
get_shape_model_contours (ModelContours_dsgjtime, ModelID_dsgjtime, 1)
for I := 0 to |Score_dsgjtime|-1 by 1
    vector_angle_to_rigid (0, 0, 0, Row_dsgjtime[I], Column_dsgjtime[I], Angle_dsgjtime[I],
HomMat2DRotate_dsgjtime)
    hom_mat2d_scale (HomMat2DRotate_dsgjtime, Scale_dsgjtime[I], Scale_dsgjtime[I], Row_dsgjtime[I],
Column_dsgjtime[I], HomMat2DScale_dsgjtime)
    affine_trans_contour_xld (ModelContours_dsgjtime, ModelTrans_dsgjtime, HomMat2DScale_dsgjtime)
    dev_display (ModelTrans_dsgjtime)
endfor
disp_message (WindowHandle, '总共匹配了' + |Score_dsgjtime| + '个符号', 'window', 12, 12, 'red', 'true')
disp_continue_message (WindowHandle, 'black', 'true')
stop()

clear_shape_model (ModelID_dsw)
clear_shape_model (ModelID_qs)
clear_shape_model (ModelID_dsgj)
clear_shape_model (ModelID_dsgjtime)
clear_shape_model (ModelID_time)
clear_shape_model (ModelID_sleep)

```

一个基于 1:1 比例的匹配的实例及详解

从 1 月份暂停了 halcon 的学习，复习考研和年终收尾工作，直到这个月才开始 halcon 的学习，正好也庆祝自己考上浙江大学的工程硕士，新的学习和工作也开始了。

今天开始总结前面学习的几个例子和分享这周刚测试的关于匹配的实例分析。

学习 Halcon 以来在论坛上的几个帖子（按学习顺序步骤排列）：

[Halcon 编程交流学习区] [图像获取的实例](#)

[Halcon 实例学习交流区] [一个用摄像头获取条形码并显示类型及解码的实例及每个步骤注解](#)

[Halcon 实例学习交流区] [Halcon 支持的颜色及其名称，共 21 种](#)

[Halcon 算子学习交流区] [一个摄像头解码二维码的实例及分步骤注解](#)

[Halcon 实例学习交流区] [一段可视化的条码扫描线的实例和注解](#)

[Halcon 编程交流学习区] [Visual Studio 10.0 设置引用 HalconDotNet.dll](#)

[Halcon 编程交流学习区] [C#导出的代码，怎么调用？](#)

[Halcon 实例学习交流区] [第一个 Blob 分析的例子，抽取 Blobs](#)

基本上比较详细得解释和学习这几个例子，发现在群上对于二维码的问题，大部分我都能理解和帮助到大家，挺欣慰。

这周开始了匹配方面的学习，有两个有用的例子和大家分享，一个是基于 1:1 比例的匹配的实例，一个是基于可变比例的匹配，正在测试的是基于变形可变比例的匹配。

这个例子是基于 1:1 比例的匹配的实例。

* 在一个图片中获取 ROI 并在此图片中匹配

```
dev_close_window ()  
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)  
* 窗口语句  
read_image(Image,'L:/Halcon test/mk2.jpg')  
*read_image(Image,'L:/Halcon test/mk3.jpg')  
*read_image(Image,'L:/Halcon test/mk4.jpg')
```

* 这里有 4 张图片，每一张都说明一个小问题，附图分析。

```
gen_rectangle1 (ROI1, 57.8333, 49.5, 181.167, 342.833)
```

* 画一个矩形选择 ROI，矩形在左上角，覆盖一个完整的，无变形规定尺寸的商标，作为模板。

```
reduce_domain(Image,ROI1,ImageReduced1)
```

* 大图和这个矩形的 ROI 相减就会得到一个左上角的商标的图案作为模板，命名 ImageReduced。

```
create_shape_model(ImageReduced1,0,0,rad(360),0,'no_pregeneration','use_polarity',40,10,ModelID1)
```

* 创建一个比例不变 (1:1) 的匹配的轮廓模型。具体参数下个帖子说明，也可见[Halcon 算子学习交流区]

Halcon 模版匹配算子解析

```
find_shape_model(Image,ModelID1,0,rad(360),0.7,13,0.5,'interpolation',0,0.9,Row,Column,Angle,Score)
```

* 寻找与模板的大小尺寸必须是一比一匹配的，只是角度的不同而已，若大小发生变化，则不能匹配

```
get_shape_model_contours(ModelContours1,ModelID1,1)
```

* 在大图中获取匹配。

```
for i := 0 to |Row|-1 by 1
```

```
vector_angle_to_rigid(0,0,0,Row,Column,Angle,HomMat2D)
```

```
affine_trans_contour_xld(ModelContours1,ContoursAffinTrans,HomMat2D)
```

```
endfor
```

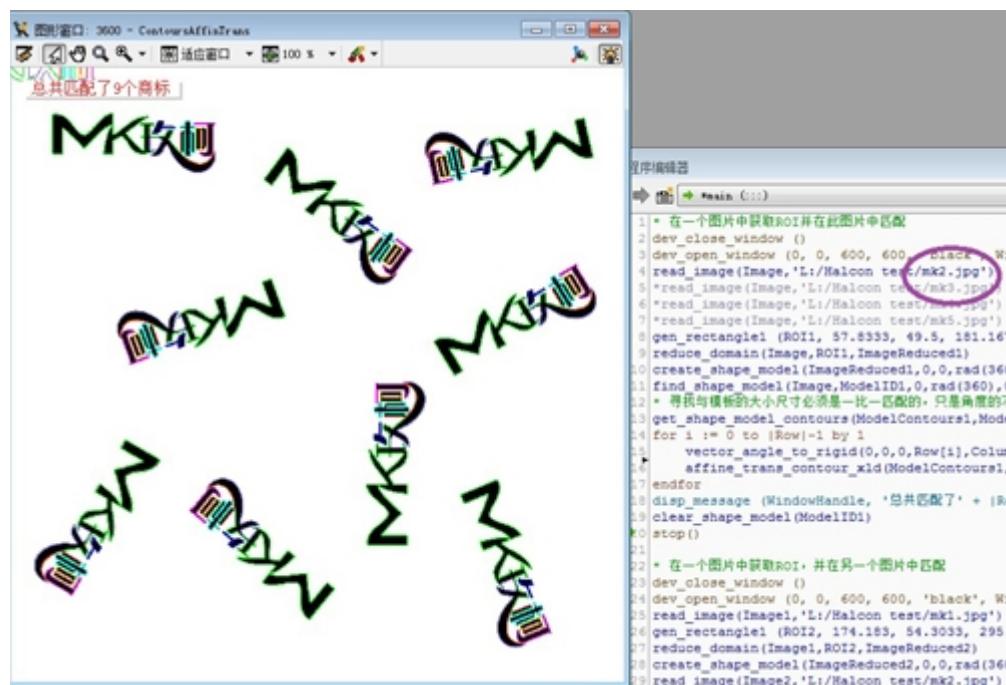
* 获取匹配。

```
disp_message (WindowHandle, '总共匹配了' + |Row| + '个商标', 'window', 12, 12, 'red', 'true')
```

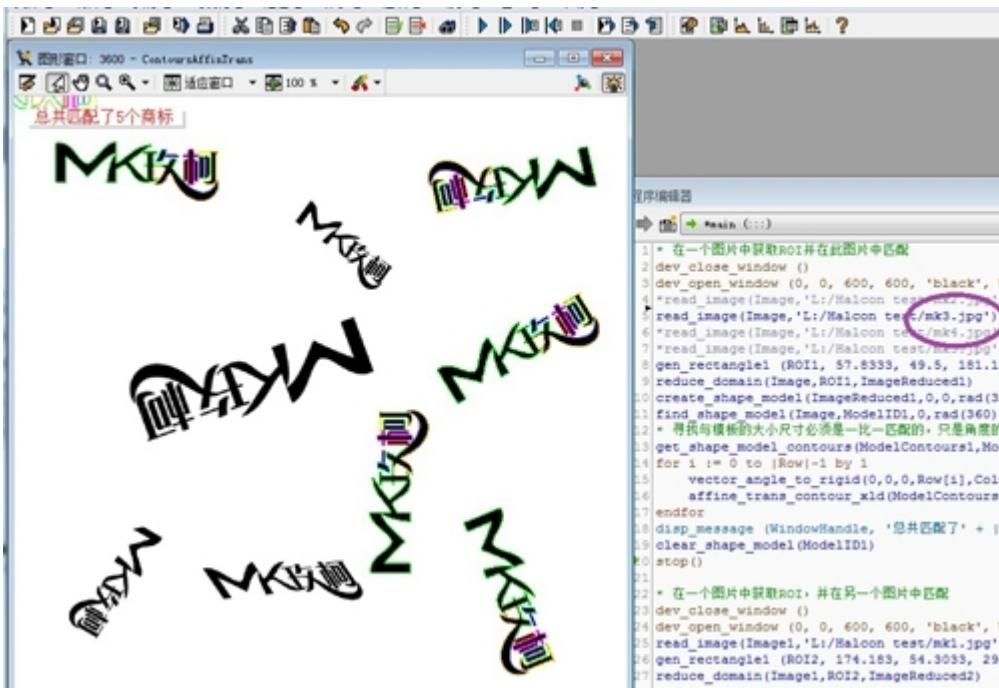
* 输出数量统计。

```
clear_shape_model(ModelID1)
```

```
stop()
```

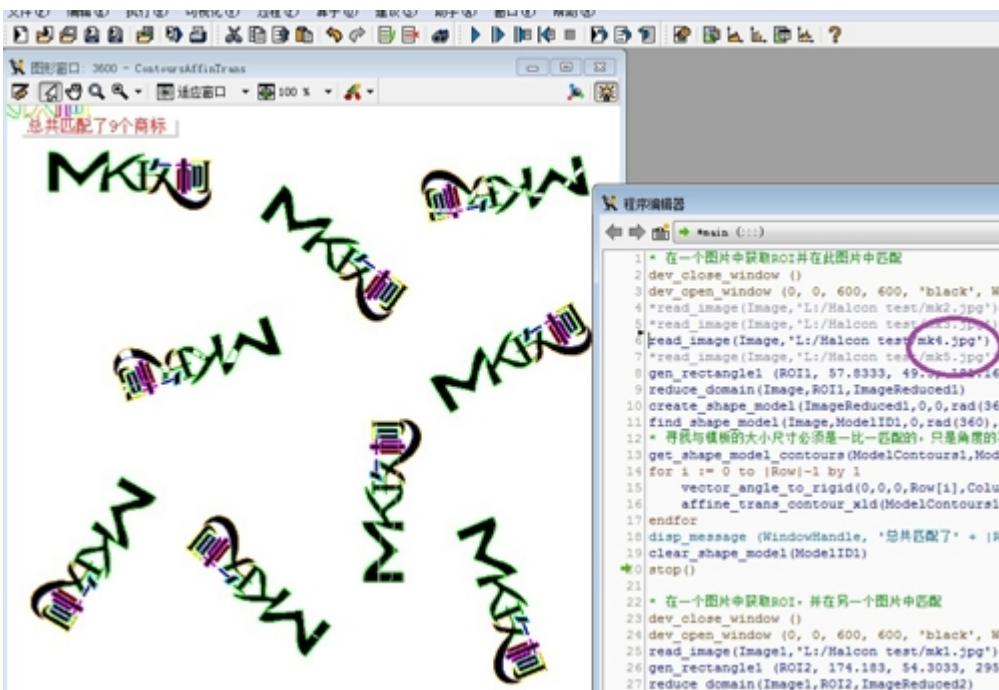


图片 mk2，共匹配出 9 个商标，商标只是角度变化（360 度，也可自己规定角度）。



图片 *mk3*, 匹配到 4 个商标, 有的商标放大, 有的缩小, 有的变形, 则不能被找出。

若要识别这类匹配, 就需要采用基于可变比例的匹配。下个帖子讲。



图片 *mk4*, 共匹配 9 个商标, 可以看到这张图里, 有些商标缺了一角, 有些商标中间多了一条白线, 等等, 商标受到不同程度的干扰认可被识别, 只要总的轮廓是一致的。

说明这个例子仍不能用于商标印刷缺陷的检测。

一个基于可变比例的匹配的实例及详解

昨天说到的有两个有用的例子和大家分享，一个是基于 1:1 比例的匹配的实例，一个是基于可变比例的匹配，正在测试的是基于变形可变比例的匹配。

《一个基于 1:1 比例的匹配的实例及详解》

今天的例子是基于可变比例的匹配

* 读取一个轮廓模型 (shape model)，使这个模型可放大或缩小地进行寻找匹配。

```
dev_close_window ()
```

```
dev_open_window (0, 0, 600, 600, 'black', WindowHandle)
```

```
read_image(Image,'L:/Halcon test/mk3.jpg')
```

* 选取一张有各种大小，角度，甚至变形的商标的图片做测试

```
*read_image(Image,'L:/Halcon test/mk5.jpg')
```

```
gen_rectangle1 (ROI, 57.8333, 49.5, 181.167, 342.833)
```

* 画一个矩形选择 ROI，矩形在左上角，覆盖一个完整的，无变形规定尺寸的商标，作为模板

```
reduce_domain(Image,ROI,ImageReduced)
```

* 大图和这个矩形的 ROI 相减就会得到一个左上角的商标的图案作为模板，命名 ImageReduced

```
inspect_shape_model (ImageReduced, ModelImages, ModelRegions, 5, 40)
```

* 创建一个对比的模型，后面两个参数：第一个是层数 (NumLevels，如 5 层)，第二个是对比度阈值（具体用法效果还不是很明白）

* 这里看到前面有位群友：[halcon8.0](#) 的一个帖子《模型匹配》提到不理解金字塔层数的意义，这里解释一下，如图：



金字塔层数在 inspect_shape_model 设置为 5 层，那么相应的在 create_scaled_shape_model 创建为 5 层，在 find_scaled_shape_model 查找为 5 层，上图中的 5 个紫色的箭头指的是就是金字塔层数，层数越多越容易被查找，反应的速度也就越快，一般是 1-10 层，但要注意保证最底层依然能被识别，主要是作为模板的图片的精度影响。一般 5 层左右识别的速度都可以接受了。

```
create_scaled_shape_model (ImageReduced, 5, rad(-180), rad(360), 0, 0.1, 5.0, 0, ['none','no_pre_generation'], 'ignore_global_polarity', 40, 10, ModelID)
```

* 创建一个比例不变的匹配的轮廓模型

```

* create_scaled_shape_model(Template :: NumLevels, AngleStart, AngleExtent, AngleStep, Scale
Min, ScaleMax, ScaleStep, Optimization, Metric, Contrast, MinContrast : ModelID)

* 01、Template,//reduce_domain 后的模板图像

* 02、NumLevels,//金字塔的层数，可设为“auto”或 0—10 的整数

* 03、AngleStart,//模板旋转的起始角度

* 04、AngleExtent,//模板旋转角度范围, >=0

* 05、AngleStep,//旋转角度的步长, >=0 and <=pi/16

* 06、ScaleMin,//模板最小比例

* 07、ScaleMax,//模板最大比例

* 08、ScaleStep,//模板比例的步长

* 09、Optimization,//设置模板优化和模板创建方法

* 10、Metric, //匹配方法设置

* 11、Contrast,//设置对比度

* 12、MinContrast,//设置最小对比度

* 13、ModelID,//输出模板句柄

* write_shape_model (ModelID, 'mk.shm')

* 把这个模型保存成一个 shm 模板

*clear_shape_model (ModelID)

stop()

*read_shape_model ('mk.shm', ModelID)

find_scaled_shape_model (Image, ModelID, rad(-180), rad(360), 0.1, 5.0, 0.5, 0, 0.5, 'least_square
s', 5, 0.8, Row, Column, Angle, Scale, Score)

* 匹配一个比例不变的轮廓模型，参数和创建的参数差不多

get_shape_model_contours (ModelContours, ModelID, 1)

for I := 0 to |Score|-1 by 1

vector_angle_to_rigid (0, 0, 0, Row, Column, Angle, HomMat2DRotate)

hom_mat2d_scale (HomMat2DRotate, Scale, Scale, Row, Column, HomMat2DScale)

affine_trans_contour_xld (ModelContours, ModelTrans, HomMat2DScale)

dev_display (ModelTrans)

endfor

disp_message (WindowHandle, '总共匹配了' + |Score| + '个商标', 'window', 12, 12, 'red', 'true')

disp_continue_message (WindowHandle, 'black', 'true')

* 但是变形则不能被识别

```

stop ()
clear_shape_model (ModelID)



总共匹配了 8 个形状相同，大小比例不同的商标，但紫色圈中的变形的商标并不能被识别。

解决 mk3 图片的放大或缩小的商标不能用这个算子识别的问题。

其实也就是更换几个算子就可实现：

```
inspect_shape_model (ImageReduced_dsgj, ModelImages_dsgj, ModelRegions_dsgj, 5, 40)
```

```
create_scaled_shape_model (ImageReduced_dsgj, 5, rad(-180), rad(360), 0, 0.1, 5.0, 0, ['none','no_pregeneration'], 'ignore_global_polarity', 40, 10, ModelID_dsgj)
```

```
find_scaled_shape_model (Image2, ModelID_qs, rad(-180), rad(360), 0.1, 5.0, 0.5, 0, 0.5, 'least_squares', 5, 0.8, Row_qs, Column_qs, Angle_qs, Scale_qs, Score_qs)
```

尺寸 1:1 的匹配，主要算子是：

- 1、create_shape_model
 - 2、find_shape_model

尺寸成比例的匹配，主要算子是：

- 1、inspect_shape_model
 - 2、create_scaled_shape_model
 - 3、find_scaled_shape_model

这两天的模板匹配的学习，发现前面两篇关于匹配的文章有一些理解错误的地方，正在做一些修正。对于关于模板匹配的几种类别，有重新的理解，也在陆续做修正。