

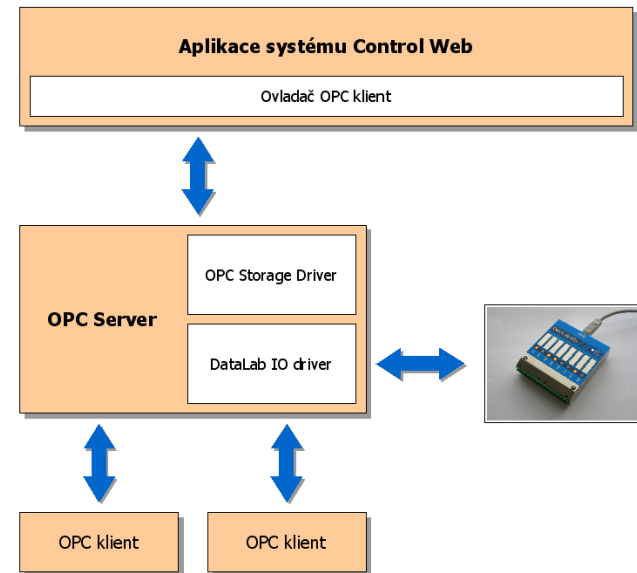
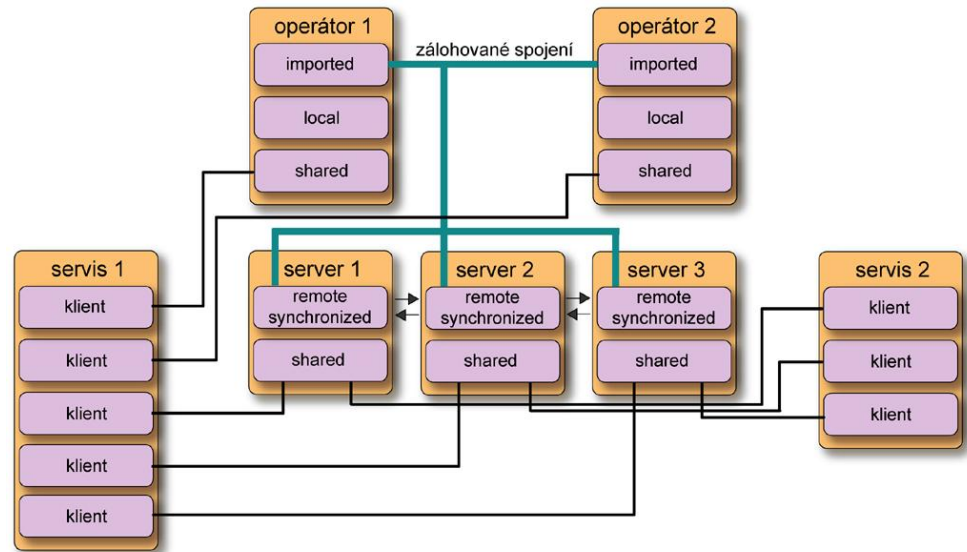
Identifikace objektů v obraze v prostředí systému VisionLab

21. 09. 2016 Liberec

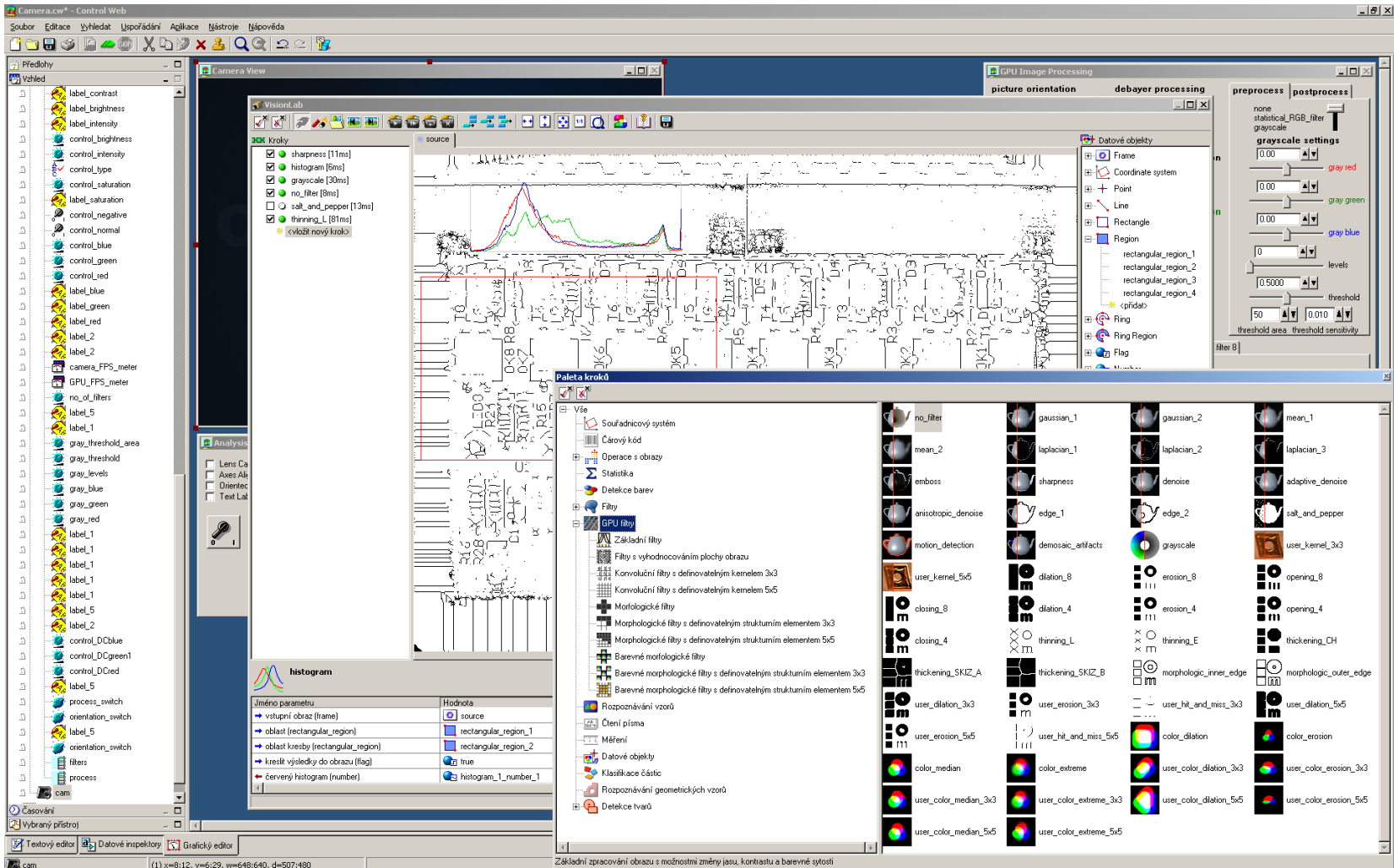
Roman Cagaš, rc@mii.cz

Control Web komponenty

- Komponenty virtuálních přístrojů
- Komponenty ovladačů
 - S vlastnických protokoly
 - Se standardními protokoly
 - Modbus, Modbus TCP
 - SMS
 - HTTP klient
 - Modem, radiomodem
 - Pro standardní sw API
 - OPC server, klient
 - Univerzální ovladače
 - TCPIP packet
 - Serial text
- Komponenty kroků strojového vidění
- Komunikace mezi jádry systému
 - Sdílení a synchronizace dat
- SQL, ODBC
- OCL – scripting nad komponentami
- Multiprocessing
- Masivně paralelní GPU processing
- Atd. atp. ...



VisionLab – systém strojového vidění



- Sada komponent instalovatelná do prostředí systému **Control Web**
- Grafický editor nad kamerovými virtuálními přístroji
- Algoritmy strojového vidění jsou zařazeny do aplikačního programu v prostředí systému **Control Web**

Příklad použití příkazového API při zpracování obrazu v kroku:

```
// begin rendering
    if (! cw_iproc::CameraCallback( CallbackHandle, vgltype::commandHIGH, (WCHAR *)& vgltype::commandLockRenderingContextW, sizeof(
vgltype::TGPUCommandData ), (ADDRESS)& GPUCommandData ) ) {
        wcsncpy( ErrorString, GPUCommandData.ErrorString.a_, ErrorString_HIGH + 1);
        ok = false;
    }
    // initialize rendering
    if (! cw_iproc::CameraCallback( CallbackHandle, vgltype::commandHIGH, (WCHAR *)& vgltype::commandInitializeRenderingW, sizeof(
vgltype::TGPUCommandData ), (ADDRESS)& GPUCommandData ) ) {
        wcsncpy( ErrorString, GPUCommandData.ErrorString.a_, ErrorString_HIGH + 1);
        ok = false;
    }
// rendering
LoadTextureFromBuffer( SourceTexture, GL_TEXTURE0, w, d, bpp, data_addr );
SetSourceDimensions( w, d, bpp );
Render( PRegion );
// copy rendered picture to output frame
    if (! cw_iproc::CameraCallback( CallbackHandle, vgltype::commandHIGH, (WCHAR *)& vgltype::commandPictureDataToOutputFrameW, sizeof(
vgltype::TGPUCommandData ), (ADDRESS)& GPUCommandData ) ) {
        wcsncpy( ErrorString, GPUCommandData.ErrorString.a_, ErrorString_HIGH + 1);
        ok = false;
    }
    // finalize rendering
    if (! cw_iproc::CameraCallback( CallbackHandle, vgltype::commandHIGH, (WCHAR *)& vgltype::commandFinalizeRenderingW, sizeof(
vgltype::TGPUCommandData ), (ADDRESS)& GPUCommandData ) ) {
        wcsncpy( ErrorString, GPUCommandData.ErrorString.a_, ErrorString_HIGH + 1);
        ok = false;
    }
// end rendering
    if (! cw_iproc::CameraCallback( CallbackHandle, vgltype::commandHIGH, (WCHAR *)& vgltype::commandUnlockRenderingContextW, sizeof(
vgltype::TGPUCommandData ), (ADDRESS)& GPUCommandData ) ) {
        wcsncpy( ErrorString, GPUCommandData.ErrorString.a_, ErrorString_HIGH + 1);
        ok = false;
    }
}
```

GPU v systémech strojového vidění

- GPU je samozřejmou součástí veškerých současných počítačů, tabletů, telefonů ...
- CPU -> jednotky výpočetních jader
- GPU -> stovky jader, široké sběrnice, vysoká datová propustnost
- VisionLab – automatický multithreading na více jádrech CPU + kroky v GPU



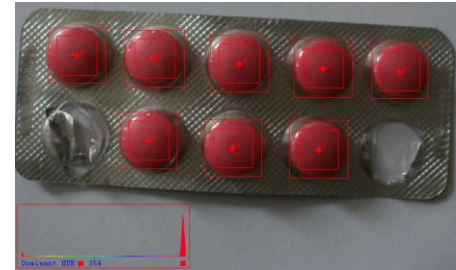
Možnosti segmentace obrazu při hledání zájmových oblastí

- Adaptivní filtrace částí obrazu, ve kterých probíhají změny



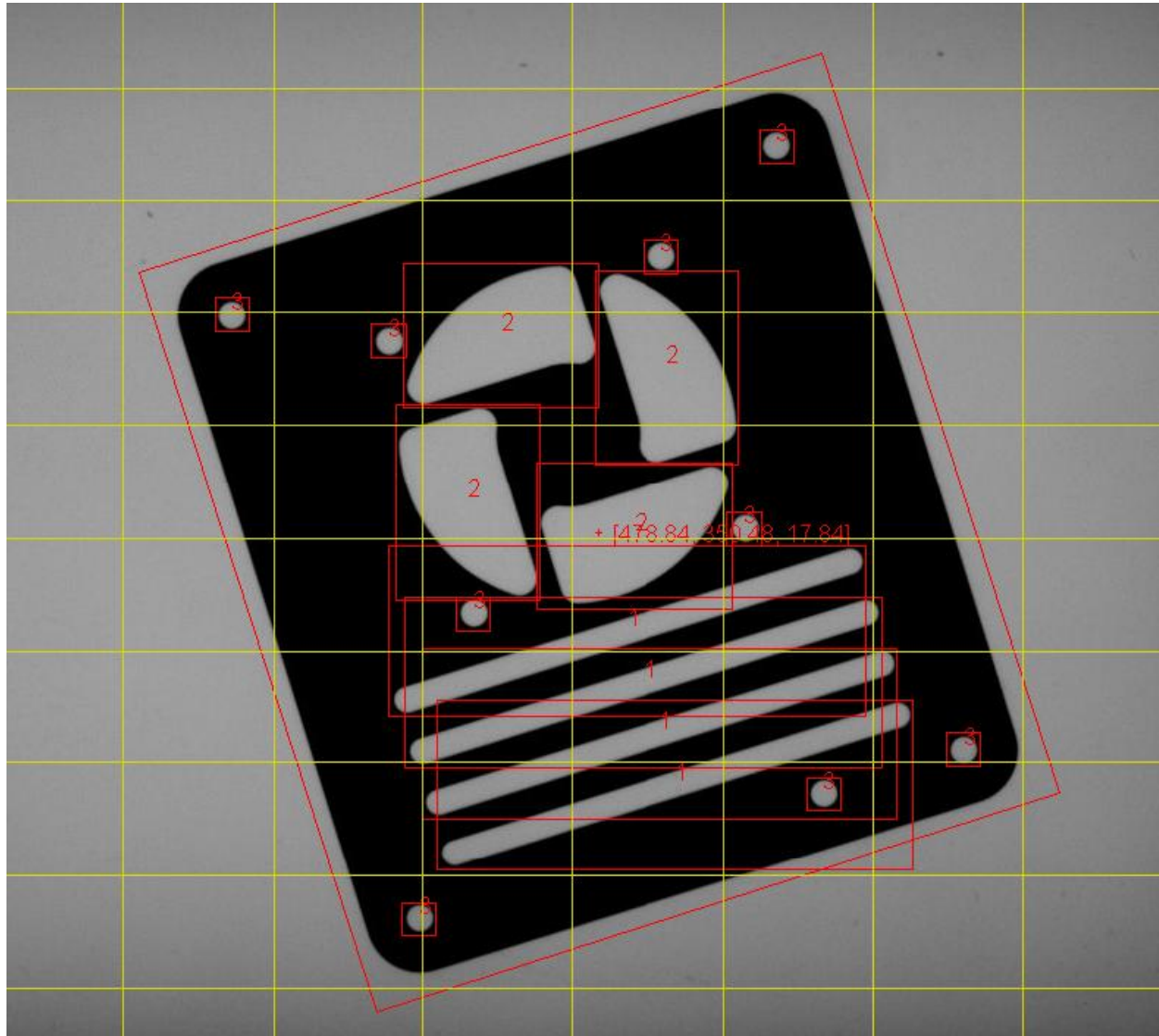
Identifikace specifických objektů

- Barvy nebo barevné patterny
- Čárové a datamatrix kódy
- Registrační značky automobilů



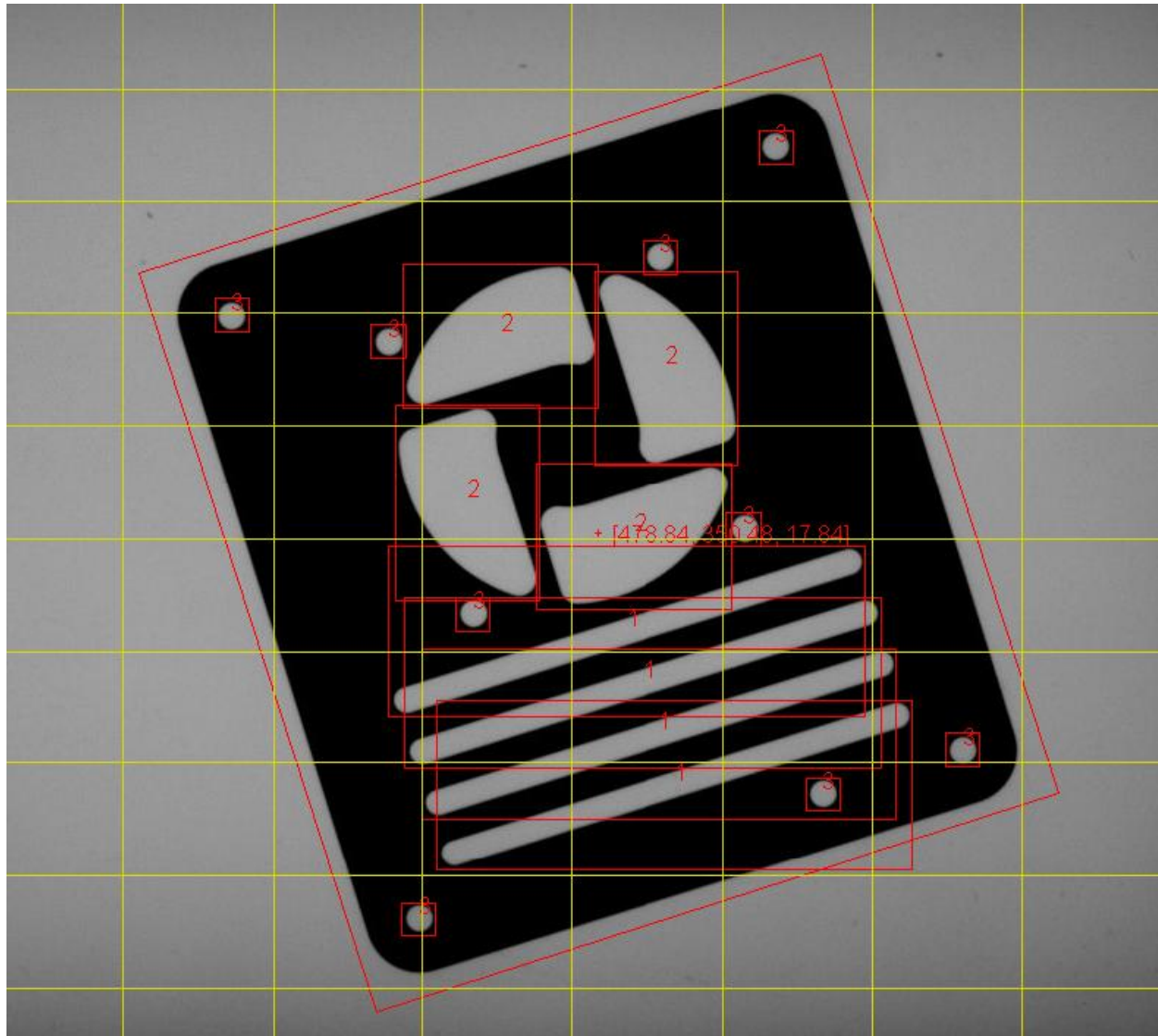
Hledání podle geometrických útvarů

- Nevýhodou je omezení na binární obraz
- Předností je přirozená invariance vůči měřítku a rotaci.



Hledání podle geometrických útvarů

- Nevýhodou je omezení na binární obraz
- Předností je přirozená invariance vůči měřítku a rotaci.



Hledání podle bitmapových vzorů

- Normalizovaná křížová korelace je náročná na výpočetní výkon
- Invariantní vůči jas, problémem je invariance vůči měřítku a rotaci
- Ani implementace v GPU nemusí mít dostatečný výkon pro větší rozsah měřítek a rotací

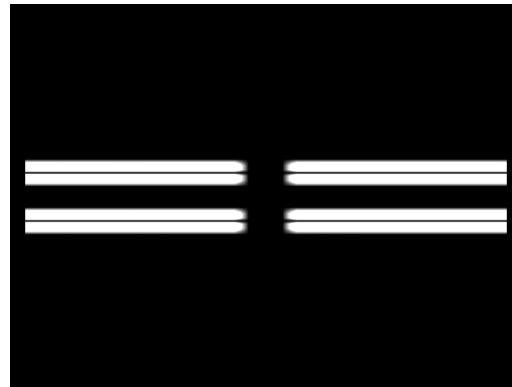
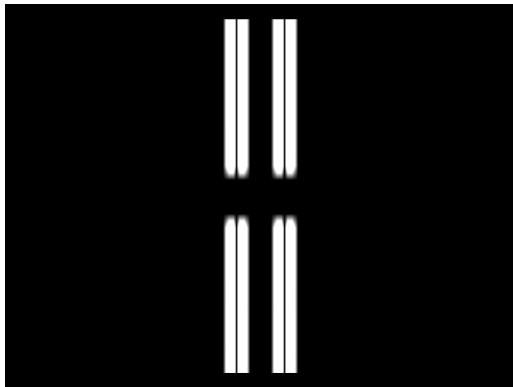
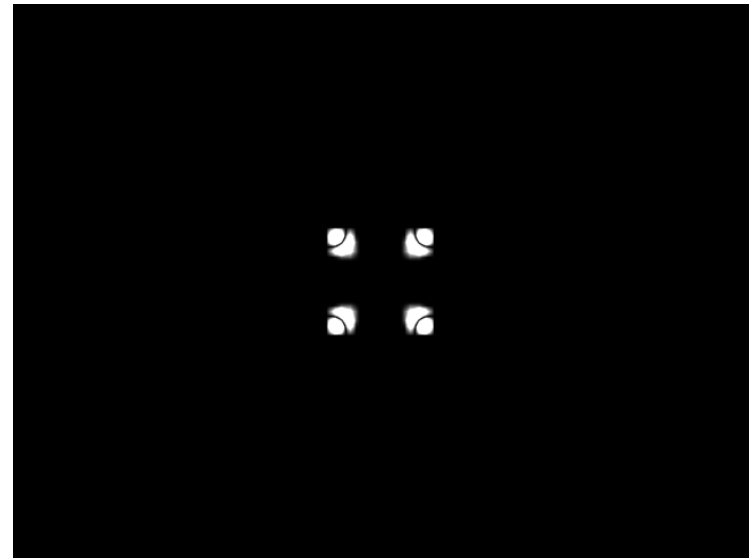
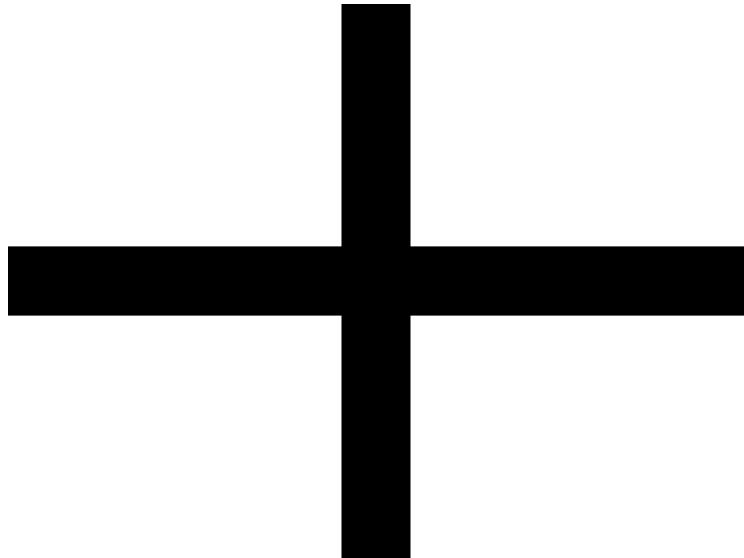
The screenshot displays the VisionLab 7.1 software interface. The main window shows a camera frame of a circuit board with several yellow circles highlighting specific components. The interface includes a menu bar, a toolbar with various tools like 'Použít', 'Čist z kamery', 'Připojení kamery', 'Nastavení světel', 'Kalibrace obrazu', 'Vyb', 'Přek', 'Provést všechny', 'Provést krok', 'Zastavit', 'Provést po vybraný', 'Opakovaně provádět', 'Automaticky po změně', 'Editor', 'Přizpůsobit', 'Přizpůsobit na šířku', 'Přizpůsobit na výšku', 'Zoom', 'Parametry editoru', and 'Nápověda'. The 'Kroky' (Steps) panel shows a sequence of steps, with 'gpu_pattern [36ms]' selected. The 'Datové objekty' (Data Objects) panel lists objects like 'Frame', 'Coordinate system', 'Point', 'Line', 'Rectangle', 'Ring', 'Flag', 'Number', 'String', and 'Color'. The 'gpu_pattern' panel at the bottom left shows a table of parameters and values.

| Jméno parametru | Hodnota |
|--|-------------|
| vstupní obraz (frame) | source |
| oblast hledání (rectangle) | rectangle_2 |
| vzorový obraz (frame) | frame_1 |
| počáteční úhel rotace (number) [-360..360] | 0 |

Below the table, there are fields for 'Soubor obrazu: pat1.jpg' and 'Soubor masky:'. The 'frame' panel shows 'frame_1' selected, with 'Počet prvků: 1' and 'Vybraný prvek: 1'. The 'Způsob inicializace' is set to 'Ze souboru'. The 'Obraz:' panel shows a small thumbnail of the highlighted component. At the bottom, the coordinates are given as 'x: 506 y: 765 r: 122 g: 119 b: 111 a: 255'.

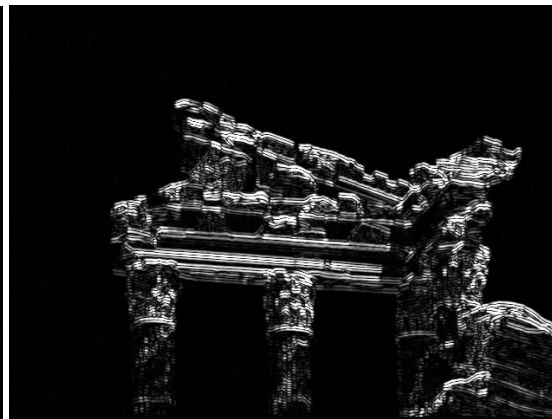
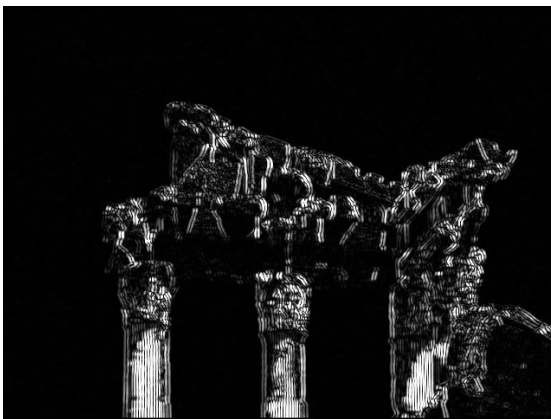
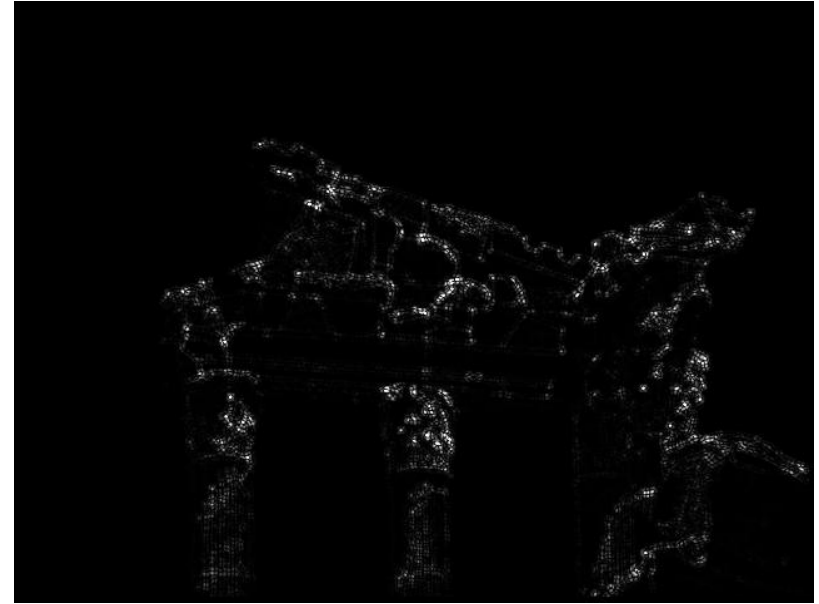
Hledání podle vlastností obrazu v okolí významných bodů

- Invariantní vůči jasů, rotaci i měřítku
- Nalezení významných bodů řešením determinantu Hessianovy matice
- Výpočet vektorů deskriptorů v okolí významných bodů



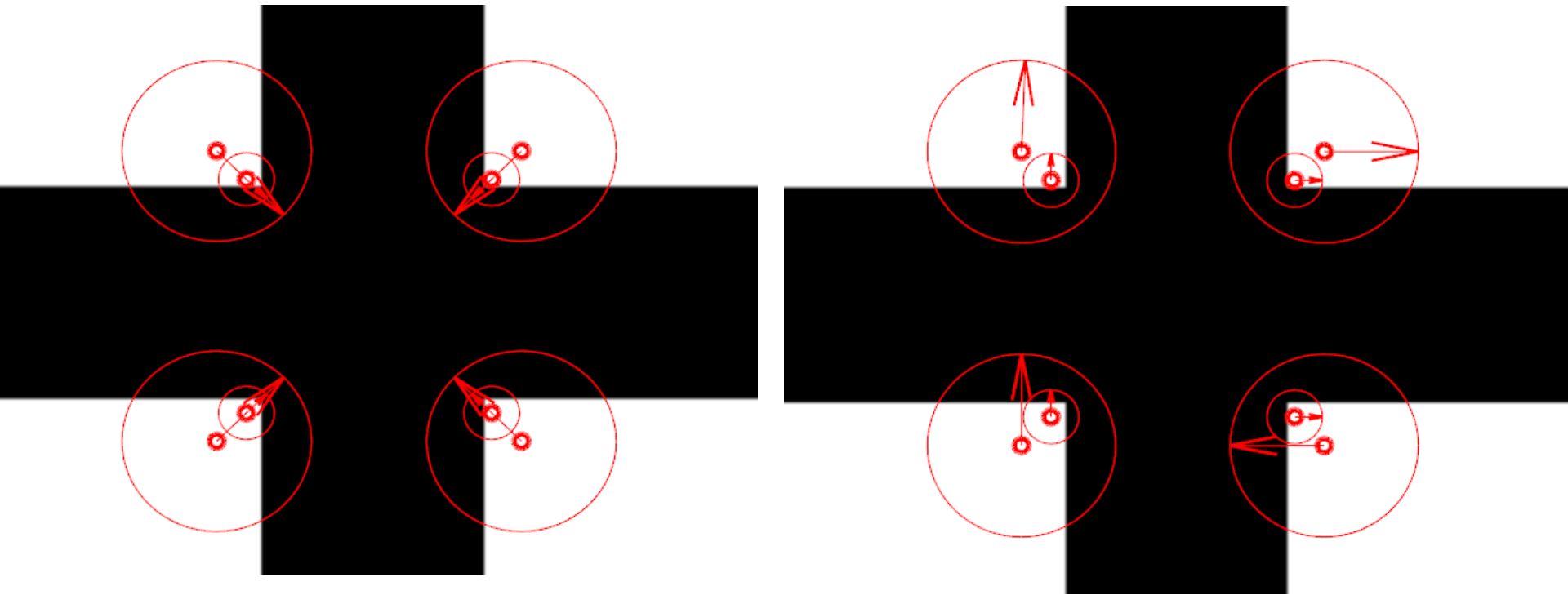
Hledání podle vlastností obrazu v okolí významných bodů

- Aproximace a diskretizace gausovských jader druhých parciálních derivací
- Haarovy vlnky, rychlý výpočet pro různá měřítka z integrálního obrazu



Hledání podle vlastností obrazu v okolí významných bodů

- Orientace významných bodů – invariance vzhledem k rotaci



Hledání podle vlastností obrazu v okolí významných bodů

- Spolehlivost nalezení i v obrazech s velkým množstvím velmi podobných oblastí

The screenshot shows the VisionLab 7.0 interface. The main window displays an image of a classical temple with a red rectangle highlighting a specific column. The 'Kroky' (Steps) panel on the left shows a workflow: 'gpu_feature_save [25ms]' and 'eval_flag [0ms]'. The 'Editor' panel at the top includes buttons for 'Přizpůsobit' (Adjust), 'Zoom', 'Parametry editoru' (Editor parameters), and 'Nápověda' (Help). Below the image, the 'gpu_feature_save' block's parameter table is visible:

| Jméno parametru | Hodnota |
|--|----------------------|
| vstupní obraz (frame) | source |
| oblast hledání (rectangle) | rectangle_1 |
| práh významných bodů (number) [0..200] | 0.1 |
| metoda zjišťování orientace významných bodů (number) | Dominantní |
| uložit deskriptory (flag) | save_flag |
| soubor s deskriptory (string) | C:\buff\features.bff |

At the bottom, a status bar indicates 'Detekce významných bodů a výpočet deskriptorů s podporou GPU' and coordinates 'x: 21 y: 23'.

This screenshot shows the same VisionLab 7.0 interface, but with the 'gpu_feature_match' block active. The main window now shows the same temple image with numerous red circles highlighting detected features. The 'Kroky' panel shows 'gpu_feature_match [47ms]'. The 'Editor' panel is identical to the previous screenshot. Below the image, the 'gpu_feature_match' block's parameter table is visible:

| Jméno parametru | Hodnota |
|--|---------------------------|
| vstupní obraz (frame) | source |
| oblast hledání (rectangle) | rectangle_1 |
| metoda zjišťování podobnosti deskriptorů (number) | Euklidovská vzdálenost |
| maximální přípustná rozdílnost deskriptorů (number) [0..200] | 1.2 |
| minimální přípustná křížová korelace deskriptorů (number...) | 0.705 |
| soubor s deskriptory (string) | c:\buff\features.bff |
| kreslit výsledky do obrazu (flag) | true |
| počet nalezených deskriptorů (number) | gpu_feature_match_0_nu... |
| počet nalezených shod (number) | gpu_feature_match_0_nu... |
| nalezený obdélník (rectangle) | FoundRectangle |
| Podmínka provedení | true |
| Výsledek | ! |

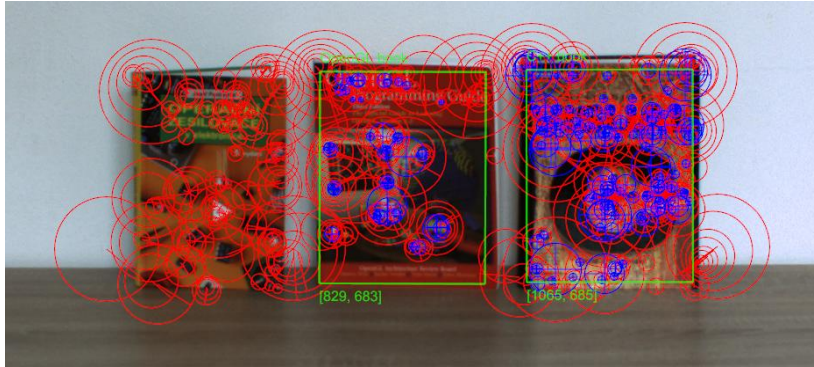
To the right of the parameter table, a 'number' block shows 'Počet prvků: 1' and 'Vybraný prvek: 1'. Below it, the 'Hodnota:' field displays '111'. The status bar at the bottom indicates 'Detekce významných bodů a porovnávání deskriptorů s podporou GPU' and coordinates 'x: 6 y: 18 r: 97 g: 122 b: 155 a: 255'.

Hledání podle vlastností obrazu v okolí významných bodů

■■■ Kroky

■■■■ : gpu_feature_match [52ms]

- Výkon a využití GPU
- Spolehlivost nalezení i v neostrých, zašuměných a barevně posunutých obrazech



VisonLab verze 7.1

Uživatelské rozhraní aplikace s menu a nástroji. Vlevo je seznam kroků, v centru hlavní obraz s detekovanými objekty, vpravo panel s parametry a tabulka s výsledky.

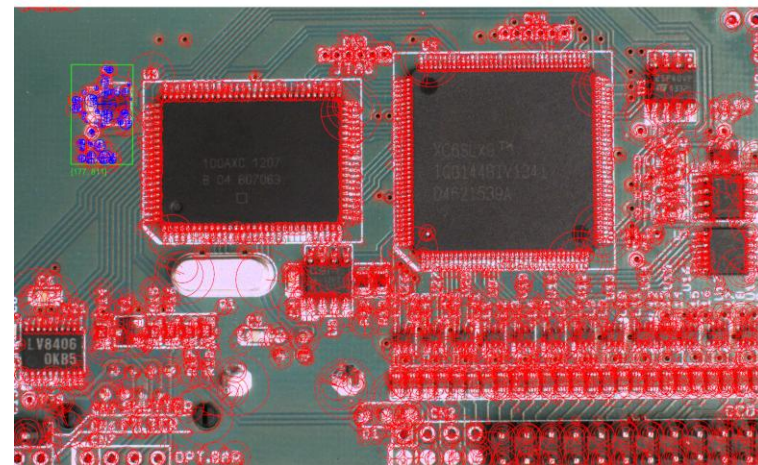
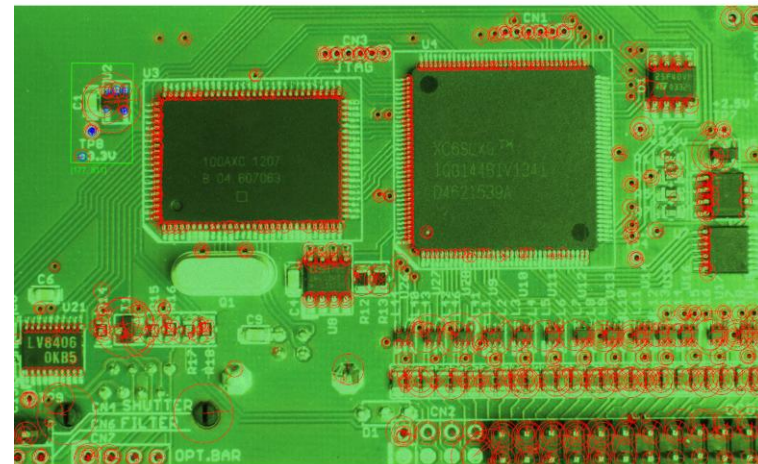
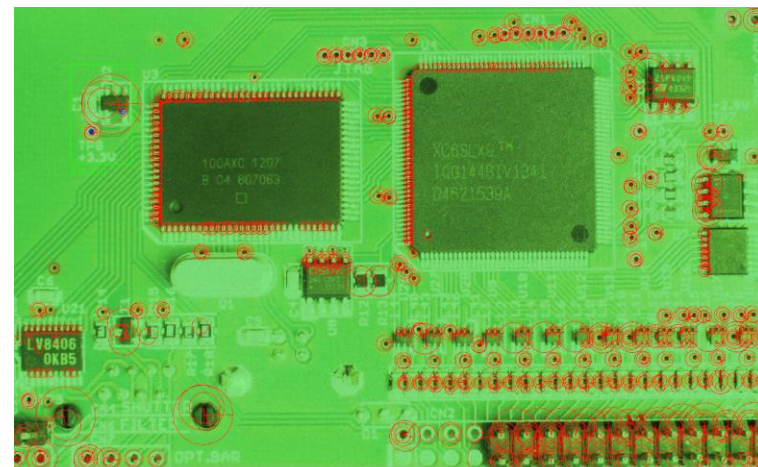
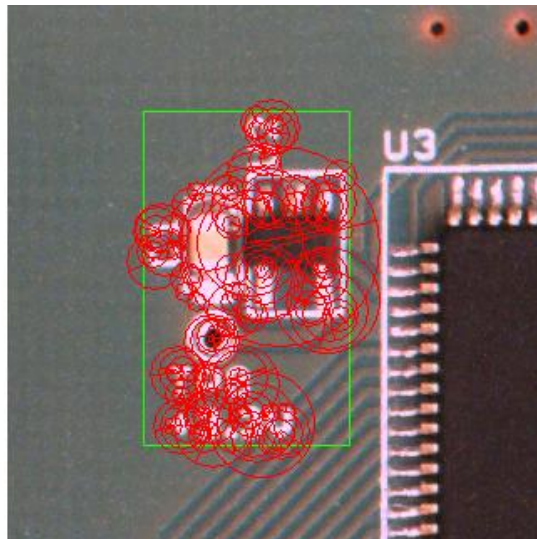
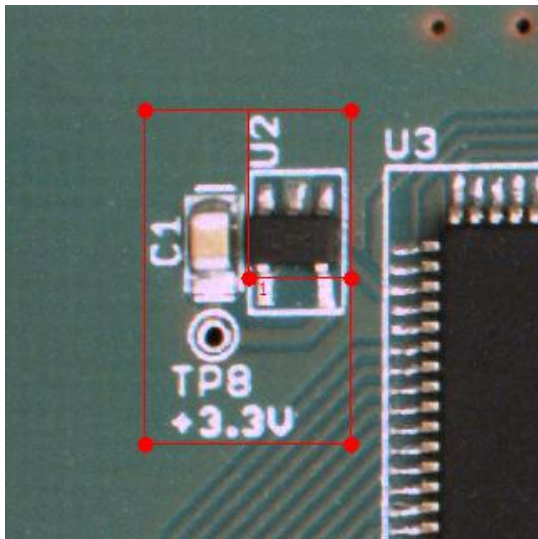
gpu_multiple_feature_match

| id | Source | Source X | Source Y | Šířka | Výška | Uhel |
|----|-----------|----------|----------|-------|----------|------|
| 1. | 1752.7794 | 8228.932 | 92 | 120 | 359.7091 | |
| 2. | 984.0286 | 1042.029 | 92 | 120 | 359.995 | |
| 3. | 662.2017 | 845.4761 | 92 | 120 | 8.471242 | |

Parametry:
- koeficient vzdálenosti od středu regionu statistické filtrace výsledků [number] [1..10]: 1.2
- úhlová odchylka regionu statistické filtrace výsledků [number] [1..60]: 0.5
- soubor s deskriptory (string): string_fles
- rozdíl vzhledu (string): string_labels
- rozsah hledání výsledků do obrazu (number): vše
- kreslit výsledky do obrazu (flag): true
- počet nalezených významných bodů (number): gpu_multiple_feature_match_0_number_1
- počet nalezených šňůr (number): gpu_multiple_feature_match_0_number_2
- nalezený obdélník (rectangle): gpu_multiple_feature_match_0_number_3
- podmínka převedení: true

Hledání podle vlastností obrazu v okolí významných bodů

- Spolehlivost nalezení ve velkém dynamickém rozsahu expozice
- Přesnost deskriptorů umožní nalezení objektu s obrazy s velmi mnoha velmi podobnými deskriptory



Hledání podle vlastností obrazu v okolí významných bodů

- Využití přesnosti detektoru pro multiměřítková hledání povrchových vad v systémech vizuální inspekce

The screenshot displays the VisionLab 7.1 software interface. The main window shows a textured image with a red circle highlighting a keypoint. The interface includes a toolbar with various tools like 'Použít', 'Uzavřít', 'Připojení kamery', 'Nastavení světel', 'Kalibrace obrazu', 'Následující', 'Provést všechny', 'Provést krok', 'Zastavit', 'Opakovaně provádět', 'Automaticky po změně', 'Přidat', 'Vložit před', 'Smazat', 'Přizpůsobit', 'Přizpůsobit na šířku', 'Přizpůsobit na výšku', 'Zoom', 'Parametry editoru', and 'Nápověda editoru'. The 'Kroky' (Steps) panel on the left shows a step named 'gpu_keypoint [104ms]'. The 'Datové objekty' (Data Objects) panel on the right shows a tree structure with objects like 'Frame', 'Coordinate system', 'Point', 'Line', 'Rectangle', 'Ring', 'Flag', 'Number', 'String', and 'Color'. The 'number' object is expanded, showing a list of 8 items with their respective values.

| # | Hodnota |
|----|---------|
| 1. | 1 |
| 2. | 0.5 |
| 3. | 0.08 |
| 4. | 0.05 |
| 5. | 0.005 |
| 6. | 0.005 |
| 7. | 0.005 |
| 8. | 0.003 |

Detekce významných bodů s podporou GPU

x: 1135 y: 964 r: 107 g: 101 b: 67 a: 255



Děkuji!

At' vám vaše systémy
strojového vidění spolehlivě
fungují a at' se při jejich vývoji
příliš nenadřete.

