ViCOS
Visual Cognitive Systems Lab

Vizualni spoznavni sistemi

Luka Čehovin

April, 2015

Laboratorij

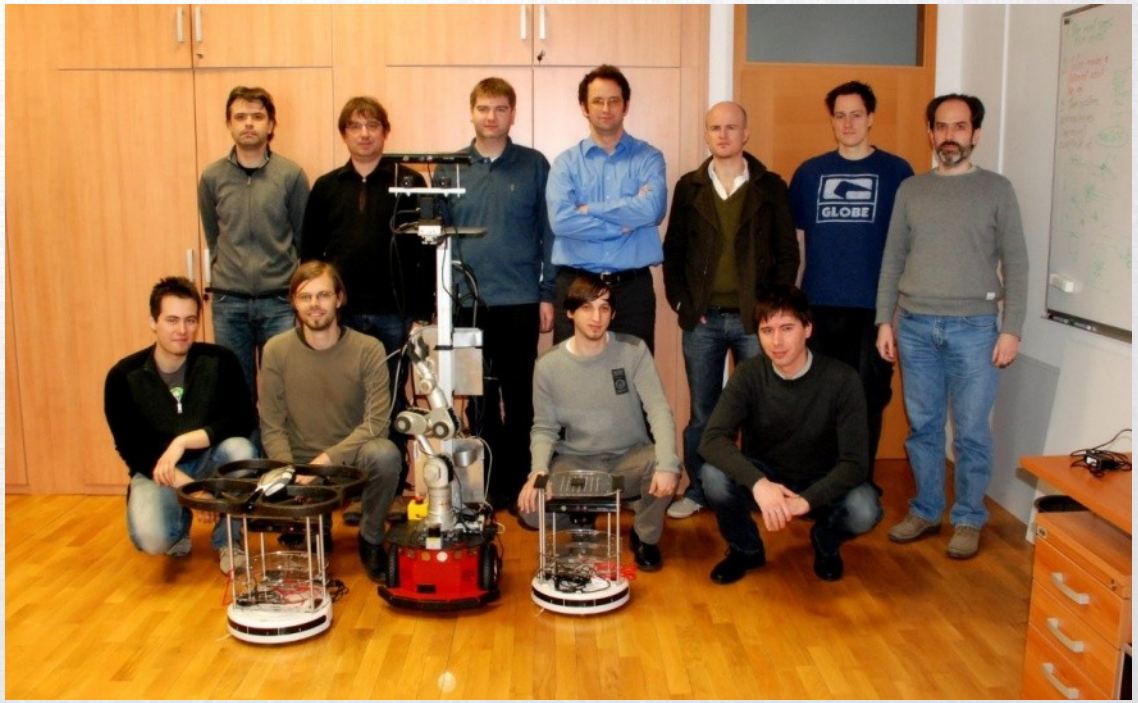
Univerza v Ljubljani,
Fakulteta za računalništvo in informatiko

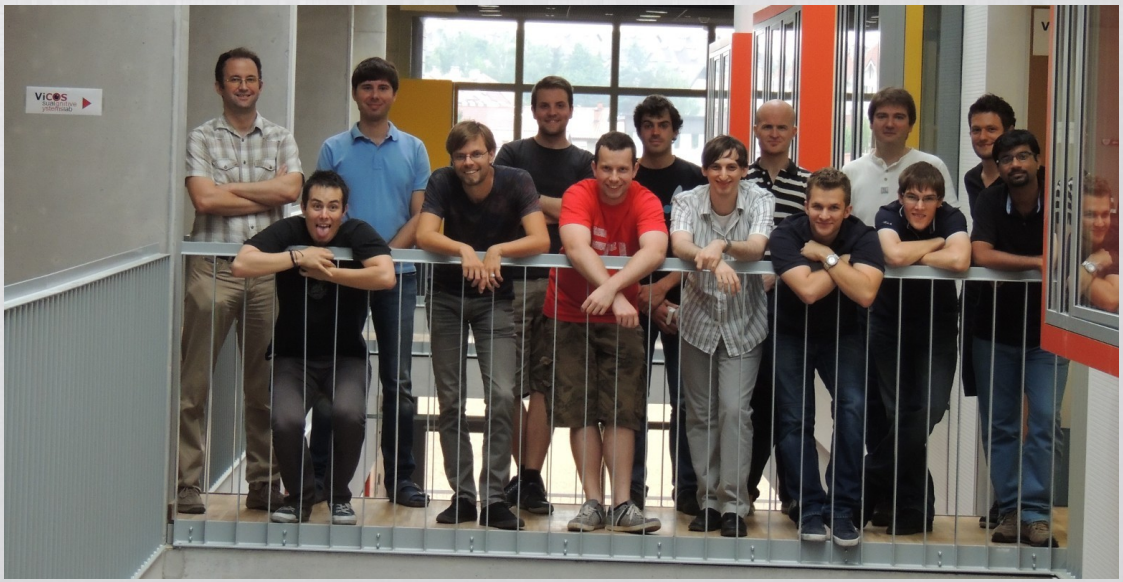
Laboratorij za umetne vizualne spoznavne sisteme

Ustanovljen leta 2004

- 7 člani
- 3 zunanji člani
- 5 študentov

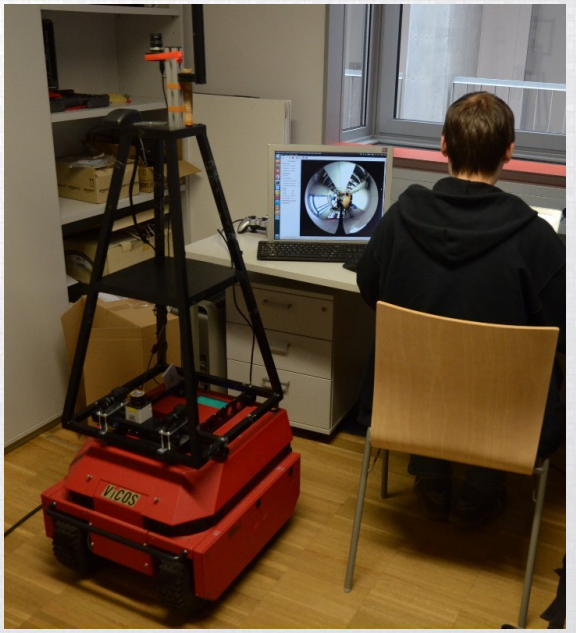
www.vicos.si

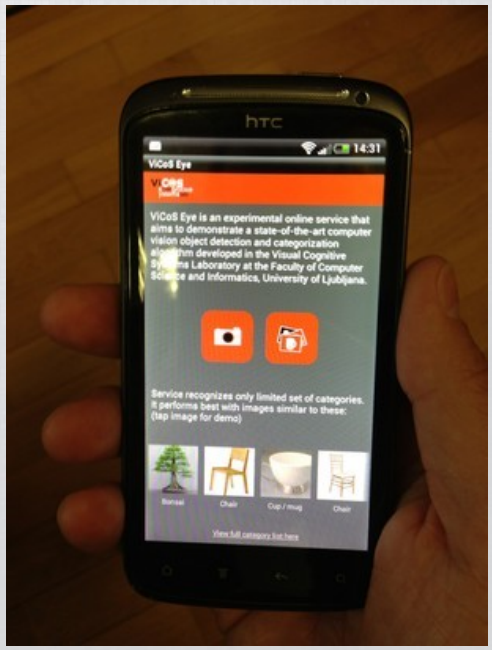




Raziskovalna področja

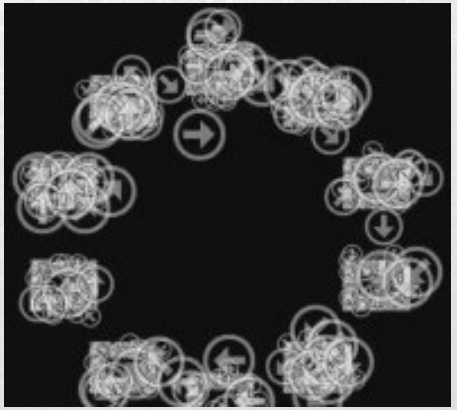
- Inteligentni vizualni sistemi
  - Računalniški vid
  - Inteligentna robotika
- Platforme
  - Mobilni roboti
  - Inteligentna okolja
  - Pametne mobilne naprave
  - Interaktivni uporabniški vmesniki

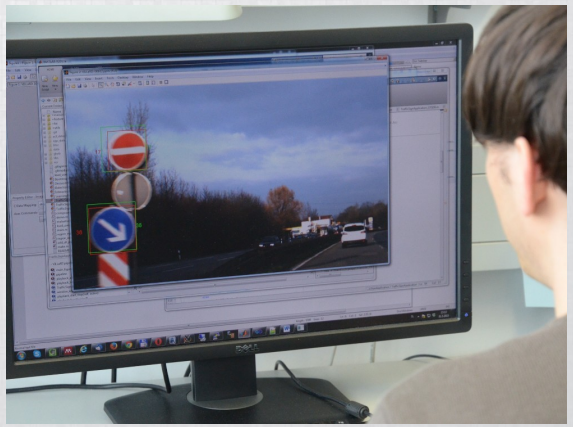


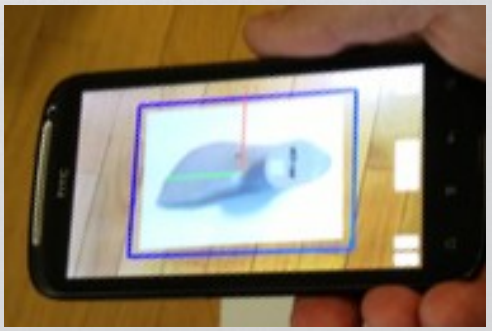


Projekti

- EU projekti
  - CoSy (2004 - 2008)
  - MobVis (2005 - 2008)
  - Poeticon (2008 - 2010)
  - CogX (2008 - 2012)
- ARRS projekti
  - LeOParts (2010 - 2013)
  - HiMoDel (2011 - 2014)
  - Villard (2014 -)
- Industrijski projekti
  - Motrr (2013 - 2014)
  - FootScApp (2013 -)







Računalniški vid

Slika

86	83	81	81	72	76	100	120	135	151	161
86	83	83	81	69	71	95	117	134	152	163
87	84	85	82	68	68	92	115	133	153	168
88	84	87	84	71	71	95	116	134	152	174
89	85	87	86	76	77	100	119	133	150	177
90	85	86	86	76	77	100	118	130	146	172
91	86	84	82	70	70	92	111	125	142	162
91	86	83	79	64	62	84	105	122	140	154
90	76	83	76	70	74	89	109	126	135	155
98	83	85	77	71	74	88	108	126	135	153
110	93	88	79	71	74	88	108	126	136	149
119	103	91	82	73	73	87	108	126	137	146
123	109	95	84	74	73	86	107	127	138	144
124	111	98	87	75	73	86	107	127	139	144
124	109	101	89	75	73	85	107	128	140	146
124	107	102	90	76	73	85	107	128	140	147
136	101	98	92	92	87	80	96	122	131	134
140	116	97	88	88	88	83	98	124	135	138
130	112	124	108	104	104	95	99	118	129	144
135	125	92	76	80	94	95	103	124	137	149

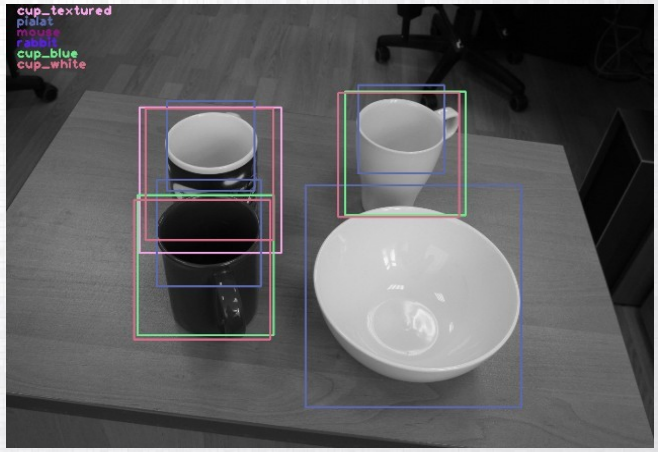
Algoritem

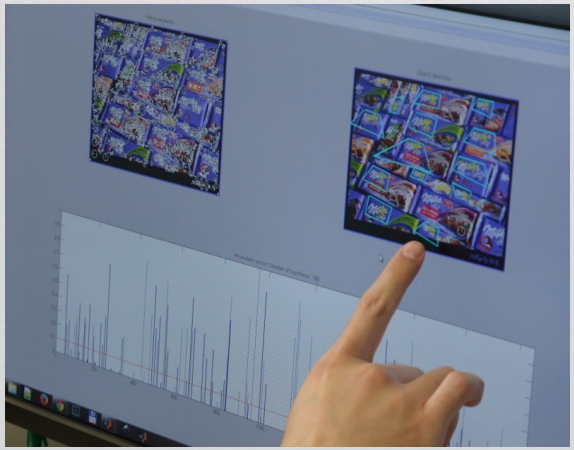
```
float best_error = FLT_MAX;
int best_consensus = 0;
Matrix3f best_model;

int* sequence = new int[patches.size()];
float* weights = new float[patches.size()];

for (int i = 0; i < patches.size(); i++) {
    weights[i] = candidates[i].size() > 0 ? patches.get_weight(i) : 0;
}

// Repeat:
for (int r = 0; r < 1000; r++) {
    random_permutation(patches.size(), sequence, weights);

    affine_from.clear();
    affine_to.clear();

    // 1. - choose three patches
    for (int n = 0; n < 3; n++) {
        affine_from.push_back(patches.get_position(sequence[n]));
        affine_to.push_back(best_candidate[sequence[n]]);
    }

    // 2. - estimate the affine transformation for positions
    Matrix3f A = compute_affine_transformation(affine_from, affine_to);

    // is model valid?

    float tt = 100;
    if (abs(A.m00) > tt || abs(A.m10) > tt || abs(A.m20) > tt
        || abs(A.m01) > tt || abs(A.m11) > tt || abs(A.m12) > tt)
        continue;

    // 3. - determine the error for the proposed model

    float error = 0;
    int consensus = 0;
    for (int n = 3; n < patches.size(); n++) {
        if (sequence[n] == -1) break;

        Point tp = transform_point(patches.get_position(sequence[n]), A);

        int mindist = INT_MAX;
```

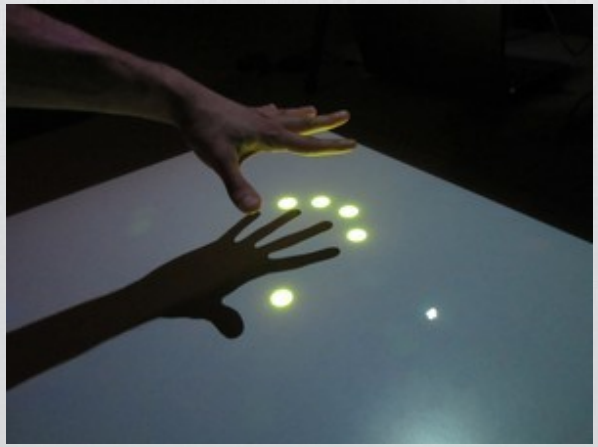
```
    for (int j = 0; j < params.elite_samples; j++) {
        Mat r = global_elite_samples.row(j);
        global_samples.row(sampled_positions[j].index).copyTo(r);
        global_elite_weights[j, 0] = sampled_positions[j].cost;
    }

    // TODO: fail?
    if (global_elite_weights(0, 0) == 0) global_elite_weights.setTo(1);

    globalC = row_weighted_covariance(global_elite_samples, global_elite_weights);

    globalM = row_weighted_mean(global_elite_samples, global_elite_weights);

    double det = globalC.at<double>(0, 0) * globalC.at<double>(1, 1) - globalC.at<double>(1, 0) * globalC.at<double>(0, 1);

#ifdef BUILD_DEBUG
    if (debug->get_zoom() > 0) {
        Mat gray = image.get_gray();
        debug->draw(gray);

        Matrix3f A = simple_affine_transformation(globalM.at<double>(0, 0), globalM.at<double>(0, 1),
            globalM.at<double>(0, 2), globalM.at<double>(0, 3), globalM.at<double>(0, 4));
        for (int j = 0; j < patches.size(); j++) {
            Point2f p = transform_point(status.get_position(j), A, center);
            debug->cross(p, COLOR_RED);
        }

        debug->text(cv::Point(3, 13), string("Global optimization, step: ") + as_string(i), COLOR_RED);

        debug->push(10);

    }
#endif

    if (det < params.terminate || samples_count >= params.max_samples) {
        for (int j = 0; j < patches.size(); j++) {
            status.converged[j, i];
        }
```

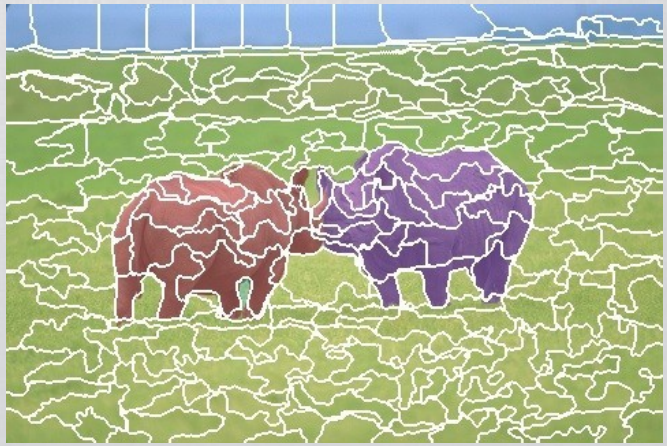
Računalniški vid

- Detekcija objektov
- Kategorizacija objektov
- Sledenje objektom
- Segmentacija
- Razpoznavanje aktivnosti
- Obogatena resničnost
- ...



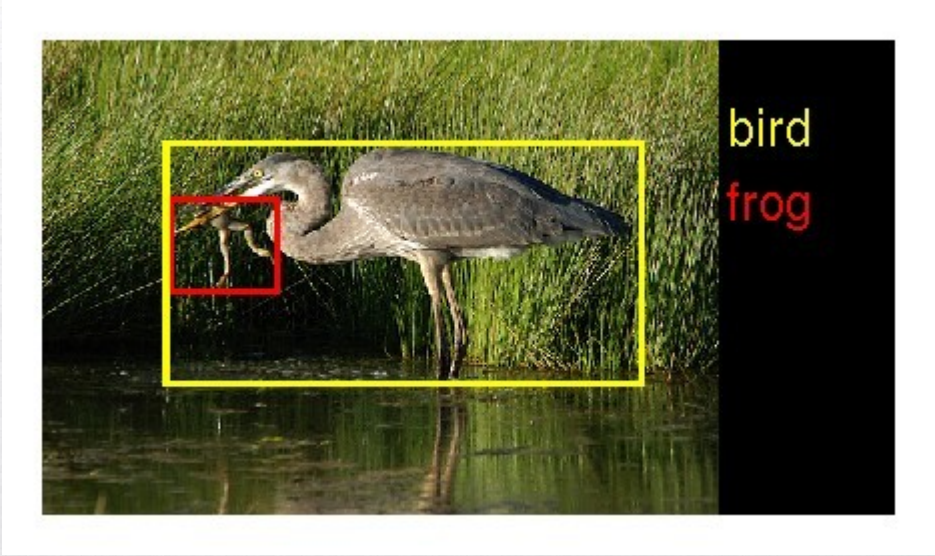




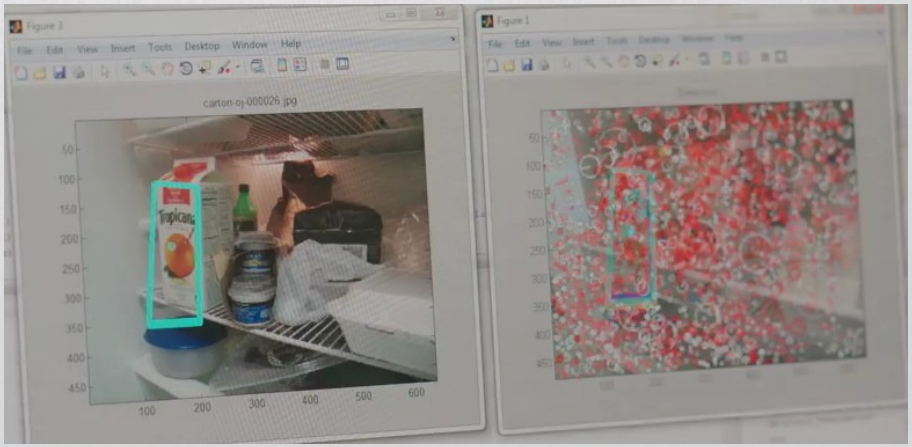


Detekcija in kategorizacija

- Kje na sliki se nahaja objekt?
- Kakšen objekt je to?

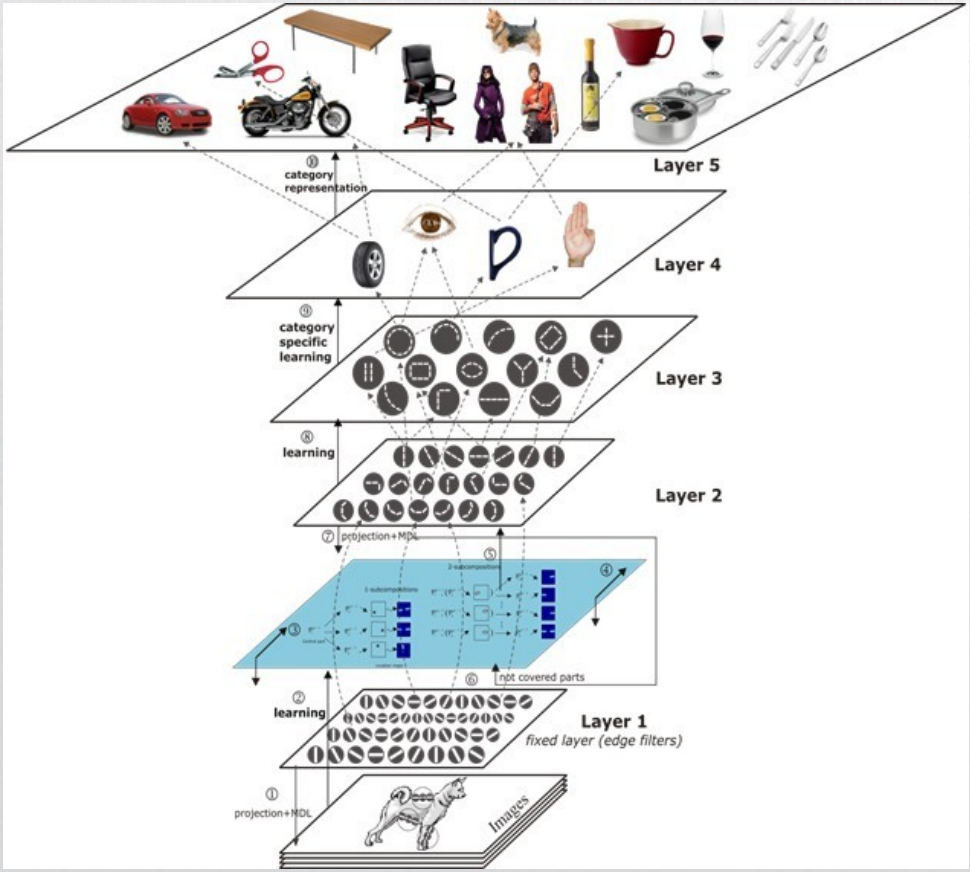


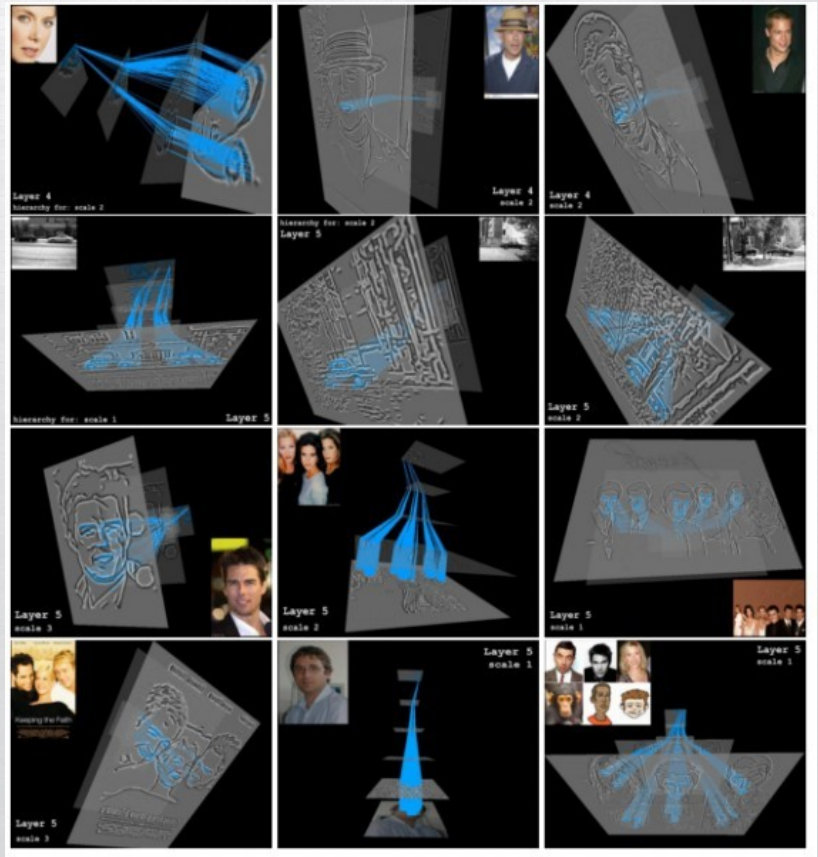
- Težavnost problema
  - Raznovrstnost izgleda
  - Zakrivanja
  - Kontekst



Naučena hierarhija delov

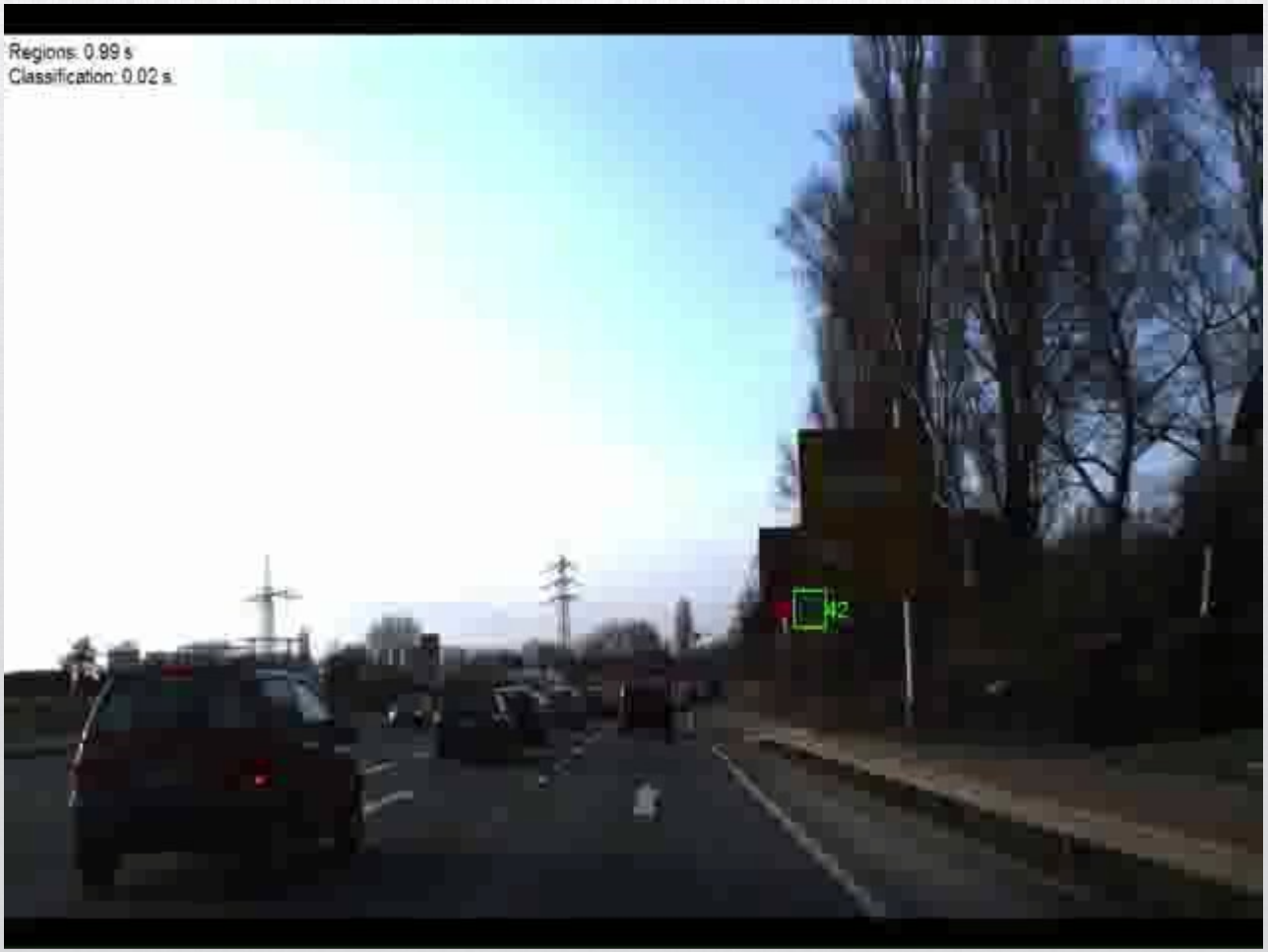
- Modeliranje sveta skozi hierarhijo delov
- Preprosti deli so deljeni med kategorijami
- Učenje novih delov z opazovanjem podatkov





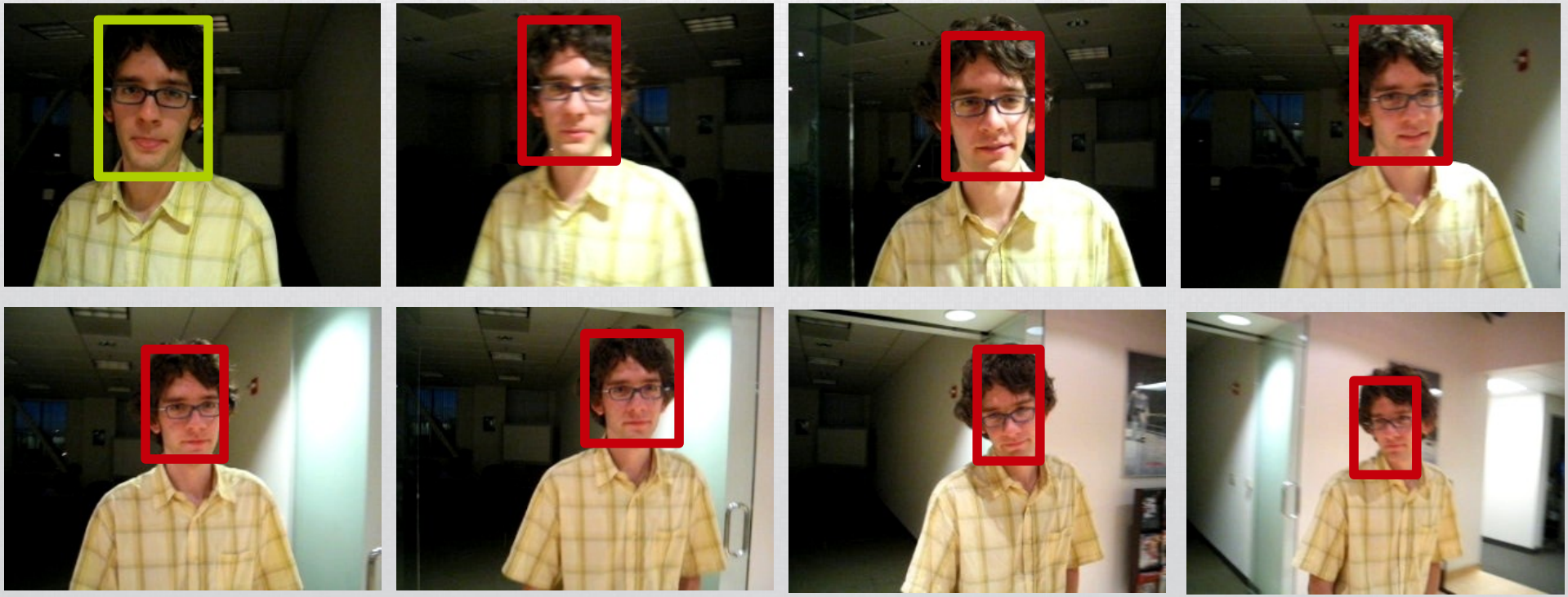
Detekcija znakov na cestah

- Vzdrževanje cestne infrastrukture
- Semi-avtomatsko prepoznavanje znakov



Vizualno sledenje

- Določanje stanja objekta v sekvenci slik
- Vizualni model – predstavitev trenutnega izgleda
- Kratkoročno vs. dolgoročno sledenje
- En objekt vs. več objektov



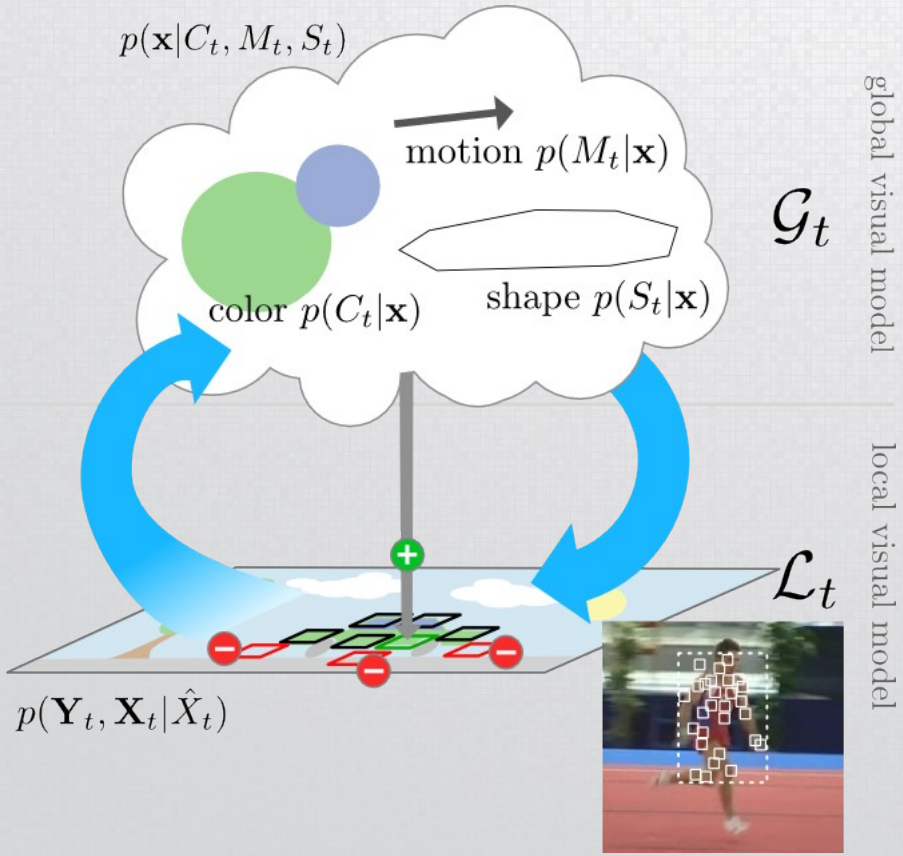
Vizualno sledenje

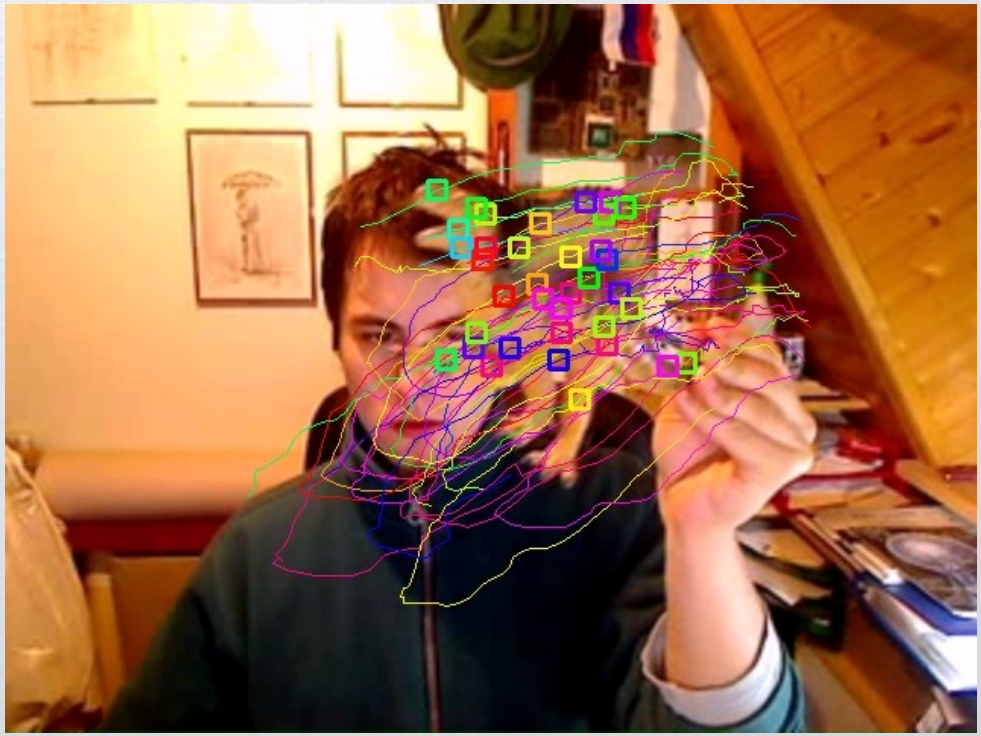
- Težavni scenariji
  - Spremembe velikosti
  - Osvetlitev
  - Artikuliranost
  - Zakrivanje

VOT
visual object tracking

Več-nivojski sledilniki

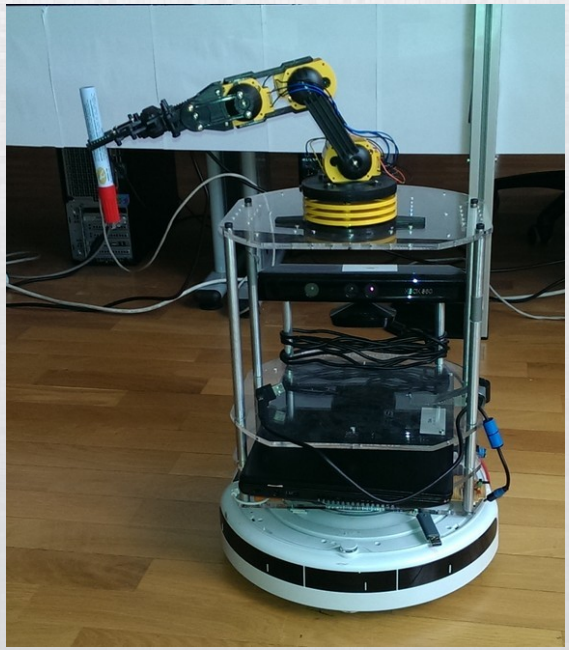
- Združevanje lokalne in globalne informacije o izgledu objekta
- Konzervativno osveževanje





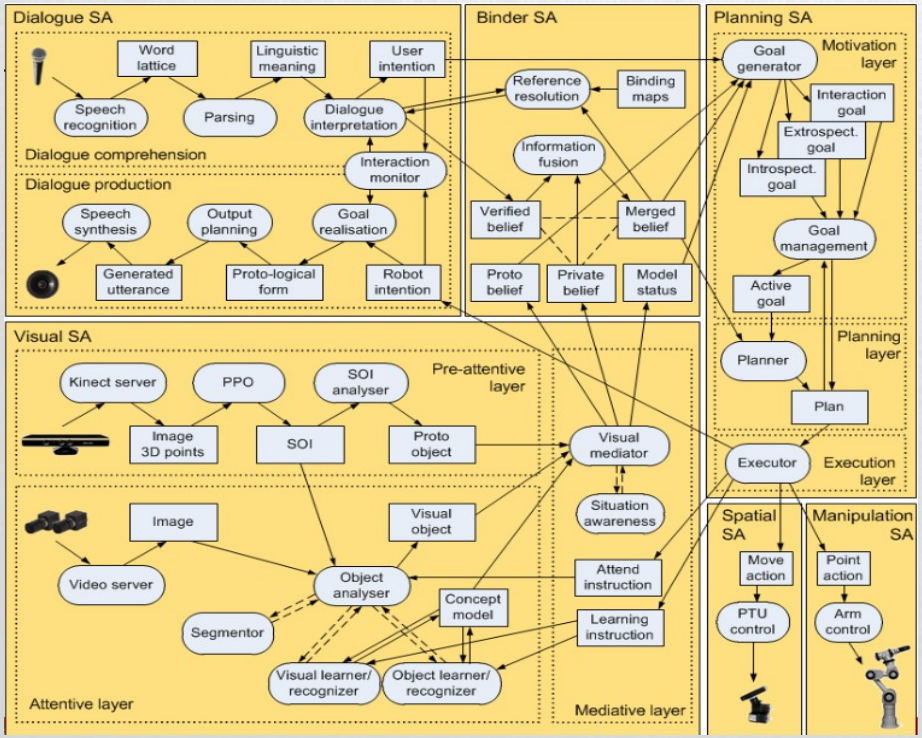
Spoznavni roboti

- Računalniški vid na mobilnih robotih
  - Interakcija s človekom
  - Navigacija



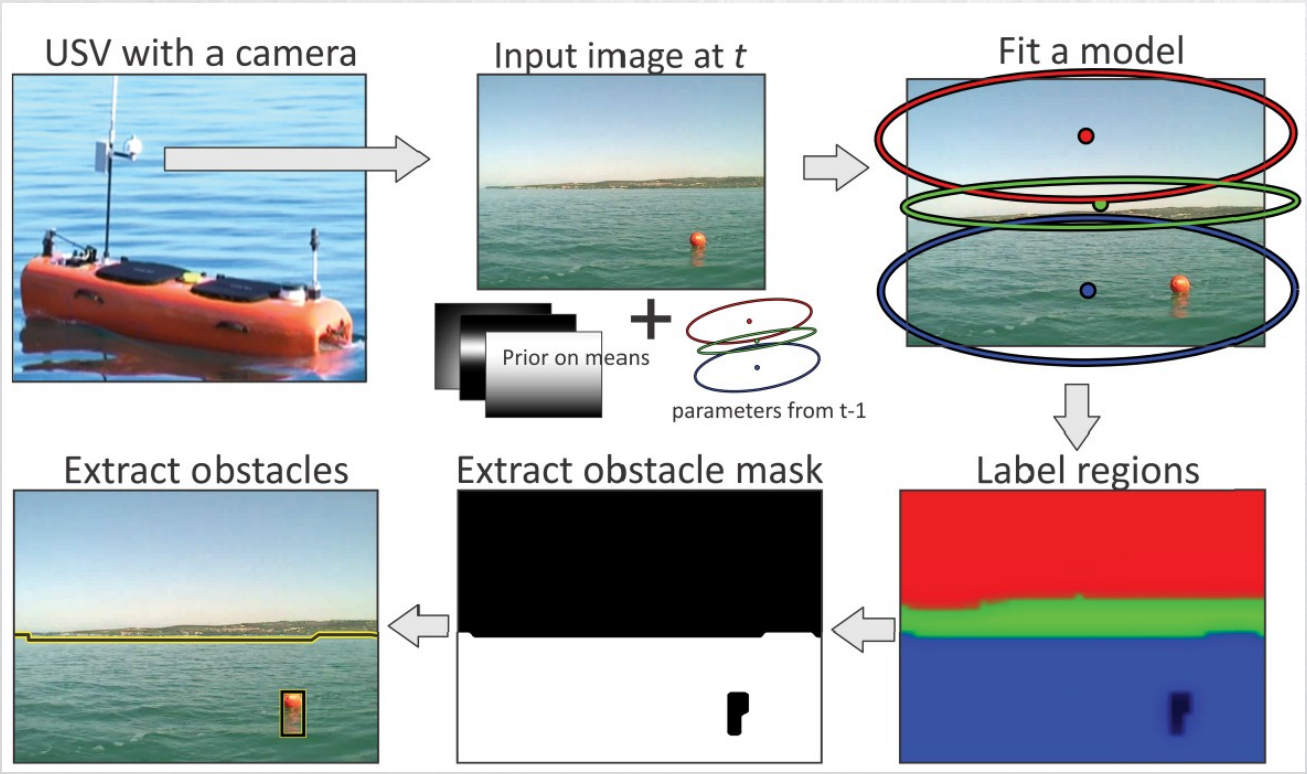
Učenje vizualnih konceptov

- Robot George (projekt CogX)
- Učenje vizualnih konceptov v dialogu s človekom



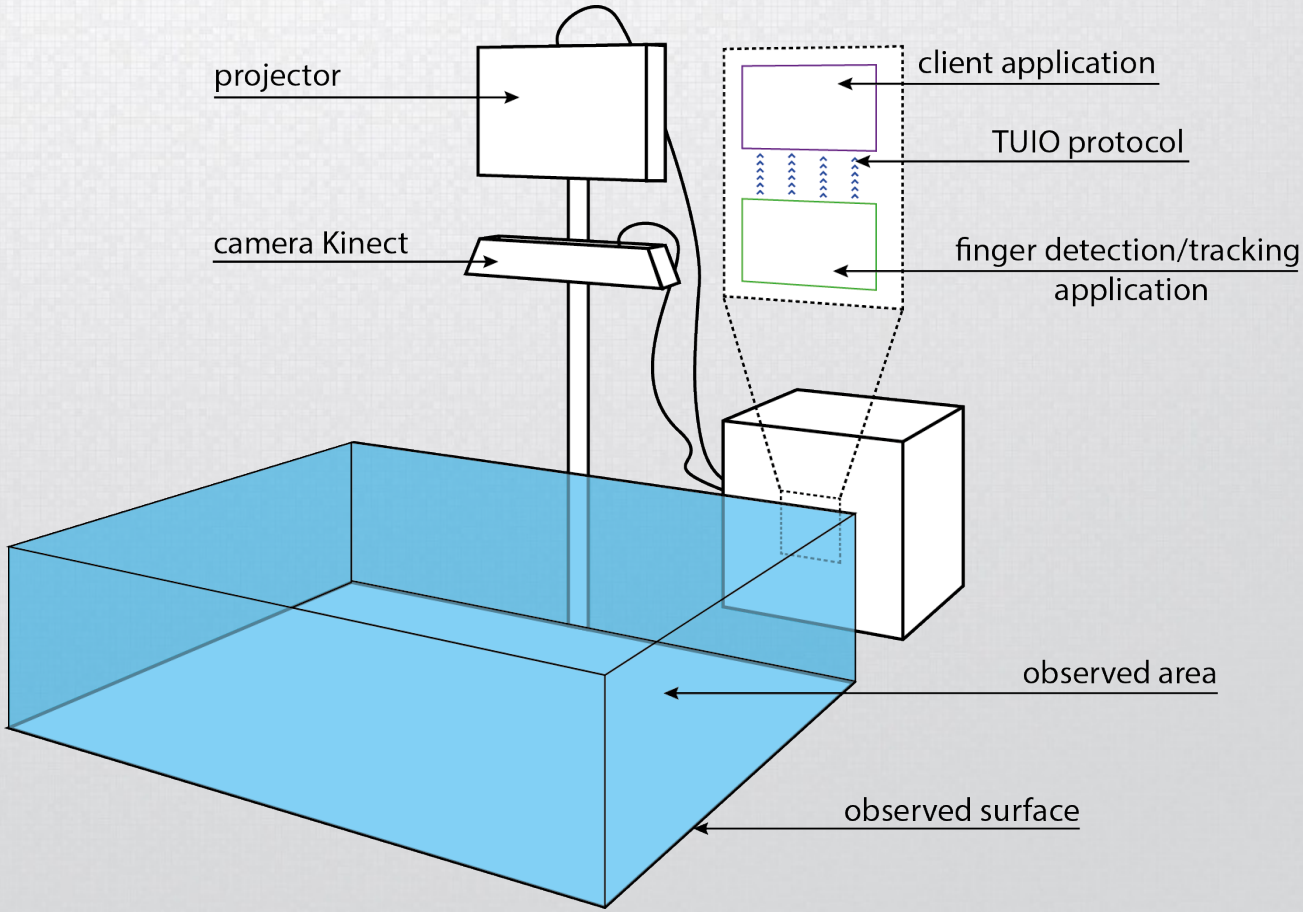
Zaznavanje ovir na vodi

- Avtonomno plovilo
- Uporaba segmentacije za detekcijo ovir



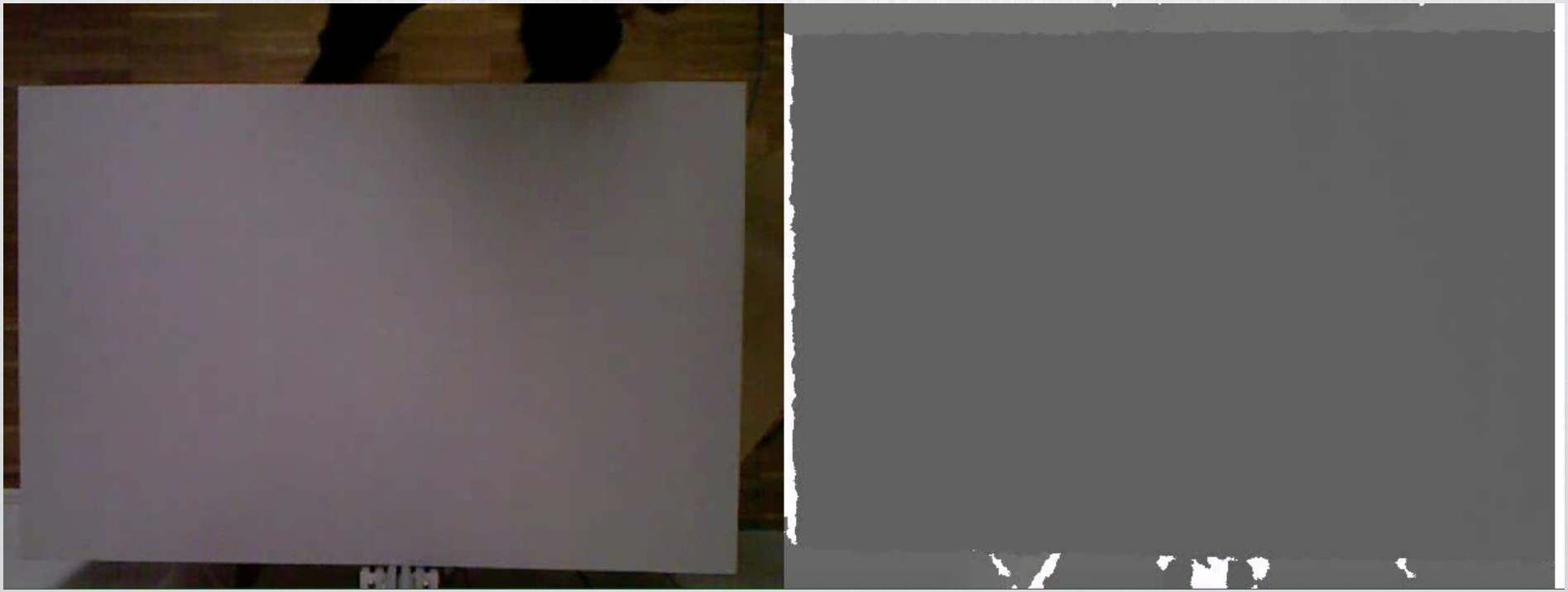
Interaktivna površina

- Napredni računalniški vmesnik
- Razvoj novih oblik interakcije



Interaktivna površina

- Avtomatska kalibracija in določitev delovne površine
- Zaznavanje konic prstov v 3D prostoru



Mobilni računalniški vid

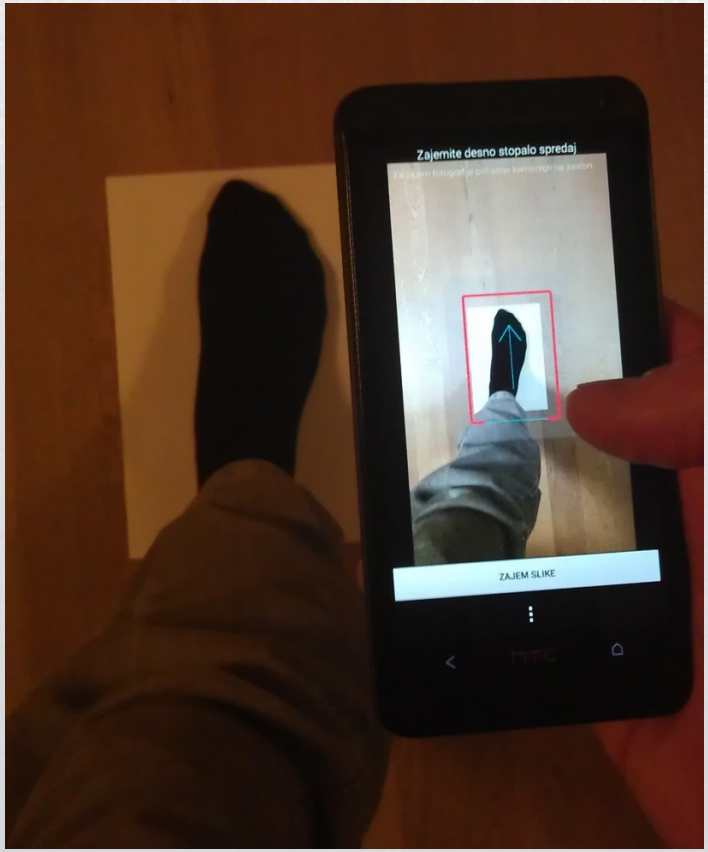
- Algoritmi računalniškega vida na mobilnih napravah
- Vgrajeni sistemi – omejena računska moč
- Izredna raznolikost strojne opreme

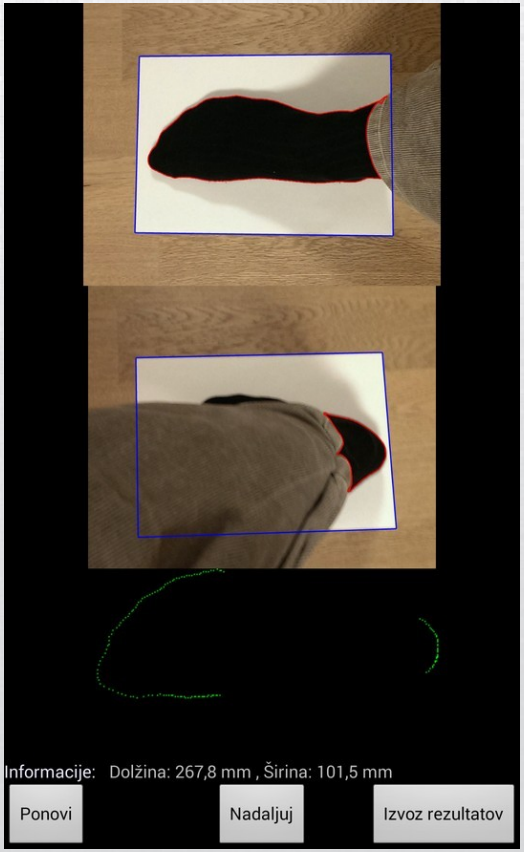
Motrr

- Sodelovanje z ameriškim startup-om
- Prenos razvitega vizualnega sledilnika na mobilno platformo

FootScApp

- Sistem za merjenje velikosti stopal z mobilnim telefonom
- Podpora raznovrstni strojni opremi





Razvijamo danes za jutri

- Zahteven razvoj
  - “Jutri” mogoče še ni jutri
  - Nekatere aplikacije mogoče že danes
  - Omejitev problema, specifikacija zahtev
  - Težko predvideti zaplete