Large Scale Particle Image Velocimetry – measuring urban discharge

Abstract

Measuring velocities and discharge in the field is well established hydroogic discipline esential for observing and studying the hydrological proceses. Small discharges formed on urban areas during rain events are usualy low-deept surface flows very dificult to measure with standard devices and procedures. New method based on image processing is introduced in hydrological practice in recent years. Large Scale Particle Image Velocimetry (LSPIV) is a non invasive method for measuring surface velocity. Since, this low cost method is easy to use in the field, it is found to be the most progressive now days.

In this article Large Scale Particle Image Velocimetry is applied in the field measurement of a urban discharge during a moderate storm event in Belgrade. The results are the velocity and the discharge. The discharge can be used as referent discharge for calibration of hydrologic model of a urban subcatchment or as an input value for sewer analisys.

Key words: LSPIV, urban discharge

Traditional Urban Discharge Measurements

In this paper the new method for field measuring of the urban discharge is introduced and applied on the example of one storm event. Urban discharge is one of the most important input parameter in any sewer system analysis, and the accuracu of this parameter have the most important impact on the results of the analysis. Traditionaly this step used to be avoided coneccting the rainfall intensity directly with the discharge in the sewer. This way there was no information about the single inflow in one single saft from a subcatchment area and the discharge coeficient of a type of soil and landuse was assumed as averaged for the whole area of subcatchment. This way a very important characteristic of a cachmant was neglected. During some havy storm events some deper flow discharges (more than 10 cm deep) use to be measured using mechanical propeller-type devices. Unfortunately, shallow water flows are especialy sensitive on local disturbtions, so this method is phisicaly imposible to apply with low error on flows deep less than 10 cm.

In recent years the Hydroacoustic Current Meter (Gary T. Fisher, P.E.1 and Scott E. Morlock2) was introduced for measuring the discharges in streams less than 10 cm (shallow streams). This non invasive method offers the continuall measurement of velocity in one spot, so the discharge is calculated using mid-section algorithm. Since the velocity profile is changing lateraly more than in the depth in shalow water flows, this method gives only approximate value of discharge in the channel.

Measuring in sewer channels are usually connected with the uncertainty of the unknown geometry and the inflow discharge from upstream. In recent years the radar systems are the most popular for estimating the surface velocity profiles, and the discharge is calculated according to choosen algorithm.

In last 20 years the new method of non invasive measuring velocity fields was introduced to the public. Regarding to computer development and the image procesing based algorithms these methods are now well established mostly in laboratory research. But, some of those algorithms were specealy adapted to be applied on the field.

One of these methods will be presented in this paper extensively and one case study is presented. This method were tested in the field during the moderate storm event on the small parking lot in front of the University building.

Non Invasive Image Processing Based Techniques For Velocity Field Measurements

During past 20 years the extensive research was concertrated on the Particle Image/Tracking Velocimetry as a non invasive image based velocity field measuring technique. These methods are based on tracking the seeded particles in the fluid velocity field for obtaininig the velocity of a fluid. The output is usualy the velocity field in two , rearly in three, dimensions, represented with the arrows that present the velocity intensity and direction.

In Particle Image Velocimetry (PIV), essence is the autocorelation or crosscorelation of two image patterns, recorded in t_1 and t_2 timesteps, and assigning the velocity according to maximum corelation coeficient registered. In preprocesing the velocity field area is divided in desired grid wich is dictates the limitations of a method-maximum and minimum velocity that can be registered. As crosscorelation method usualy 2D FFT is used, and the grid size depends on several features

In this paper the two corelation methods were compared: classic one and the 2D FFT, and several grid sizes.

Particle Tracking Velocimetry (PTV) is based on finding the path of a single particle. This method is based on otimization of two frames comparison so the picture looks the most convinient (no quick changes of velocity in the particle neibourghood).

Limitations of these two methods are that these methods are only extensively tested in the laboratory environment, and use of these methods in the field environment is only in start. The extension of PIV and PTV techniques are the Large Scale PIV (PTV) measuring techniques mostly used on river flow analisys. For the first time it is tried on shallow water of urban discharge during rain ivents.

LSPIV

The Large Scale PIV (LSPIV) is usualy used for river surface velocity profiles estimation. The procedure is:

1) seed the surface of the flow with light, ecologic, and non transparent (sugested with a great constrast to the flow) material,
2) capturing two (or more) frames on defined timestep with natural and artifitial light,

3) divide these frames to grid,
4) compare the crosscorelation coeficients,
5) use post processing techniques to reduce the errors in the obtained velocity field,
6) calculate the discharge.

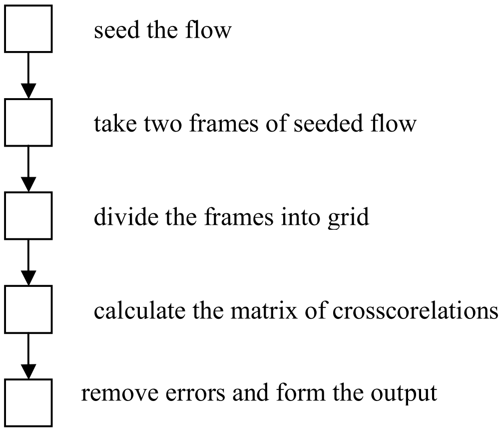
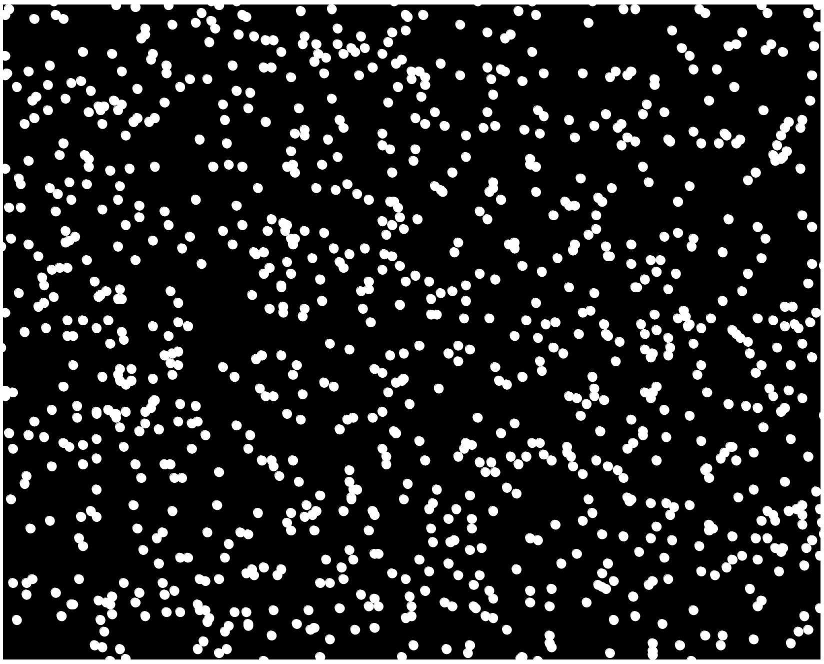
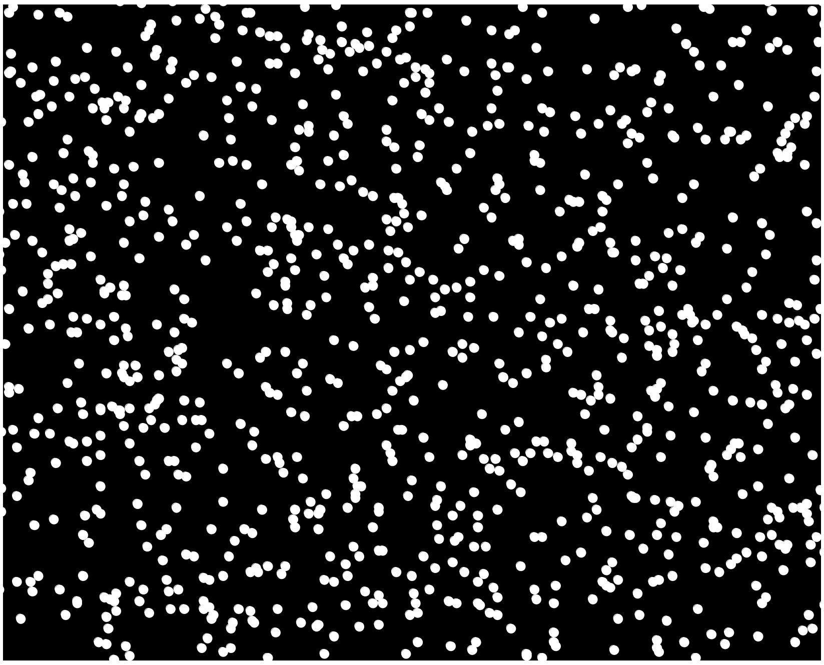


Figure 1: …

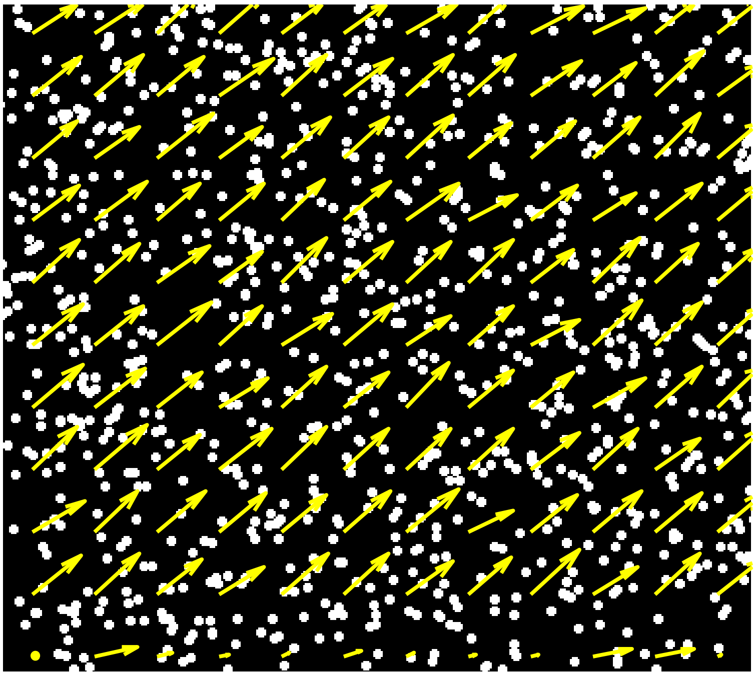
Synthesised images

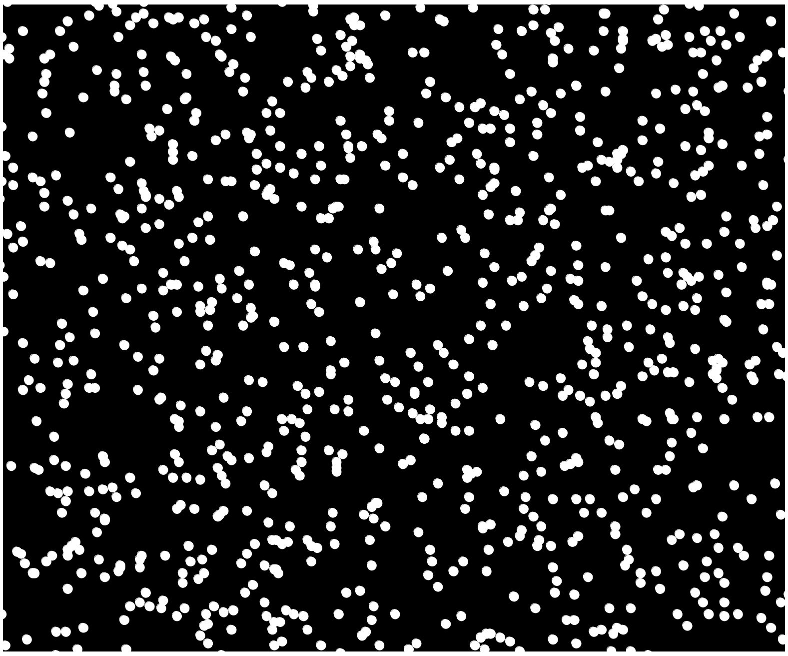
Two sinthesised images are generated with particles moved for controled displasement of u=0.5 and v=0.5:



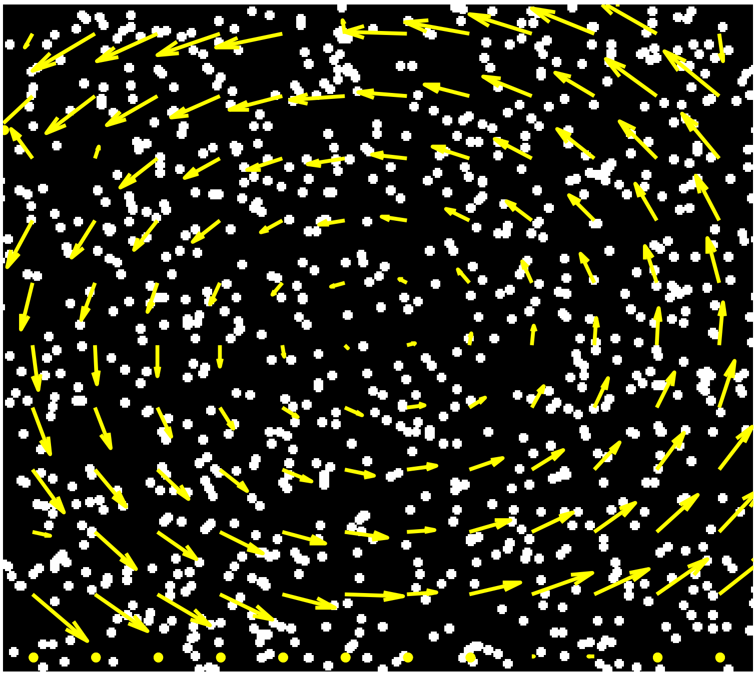


number of particles are 1024, and the size of image is 384x352 pixels.

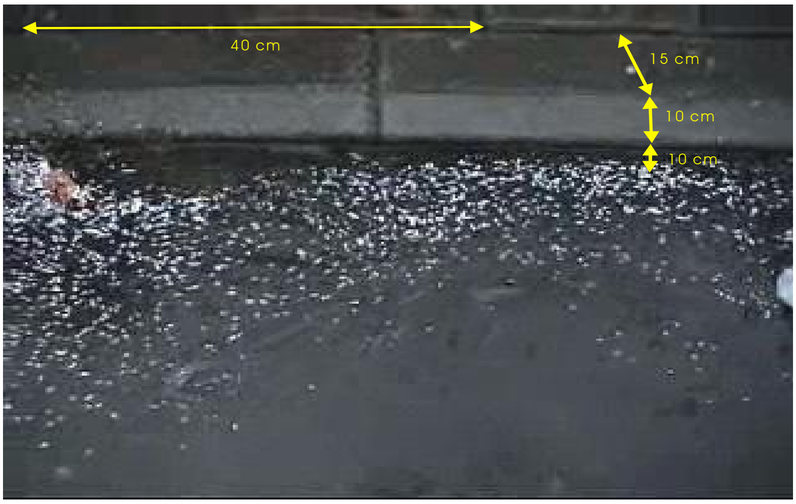


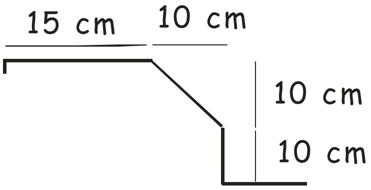


this image is turned arround center for π/35



Dimensions of pavement





Seeding

The key questions about seeding is how much the seeding should be dense? The source dencity N_s is defined by:

$$N_s = A\frac{C}{M^2}$$

where A - image area, C - concentration of particles, size of particles and M - the optic device magnification.
One frame is investigated extensively to find what is the source density of a seeding.

```
I =
imread('C:\MATLAB71\work\urban_discharge\kisa\konf_28\konf_009_0002.jpg
');
I=rgb2gray(I);

I_investigated=I(80:111,150:181);
imshow(I_investigated)
```

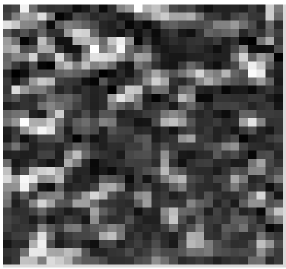


Figure …: A part of a frame32x32 pixels

The density is calculated using these assumptions:

1. The image area is A=32x32=1024 pixels
2. Concentration of trasers is estimated using image processing – calculating number of pixels with value above certain treshold value, and dividing it with the size of one particle which is estimated to be about 3x3=9 pixels.
3. Magnification of optic device is assumed to be 1.

The treshhold value is calibrated with the concentration calculated by direct counting the particles in the interogation area. Counted number of particles is 52-59, and it is assumed that one particle takes 9 pixels:

```
treshhold=65;

[i,j] = find(I_investigated>treshhold) ;

number_of_particles=length(i)/9;%should be in the range of counter particles

C=number_of_particles/(32*32);
```

Trashold is etimated to be 65, and this value depends on ilumination of a image. Now after the treshold value is calculated, the whole picture is analised:

```
treshhold=65;

int_resolution=32;
C=zeros(min(size(I)),max(size(I)));

for
ii=1:int_resolution:int_resolution*floor(min(size(I))/int_resolution)
for
jj=1:int_resolution:int_resolution*floor(max(size(I))/int_resolution)

[i,j] = find(I(ii:ii+int_resolution-1,jj:jj+int_resolution-
1)>treshhold) ;

number_of_particles=length(i)/9;
C(ii:ii+int_resolution-1,jj:jj+int_resolution-
1)=number_of_particles*ones(32,32);
end
end

cmap = contrast(C);

image(C);
```

```
colormap(cmap);
```

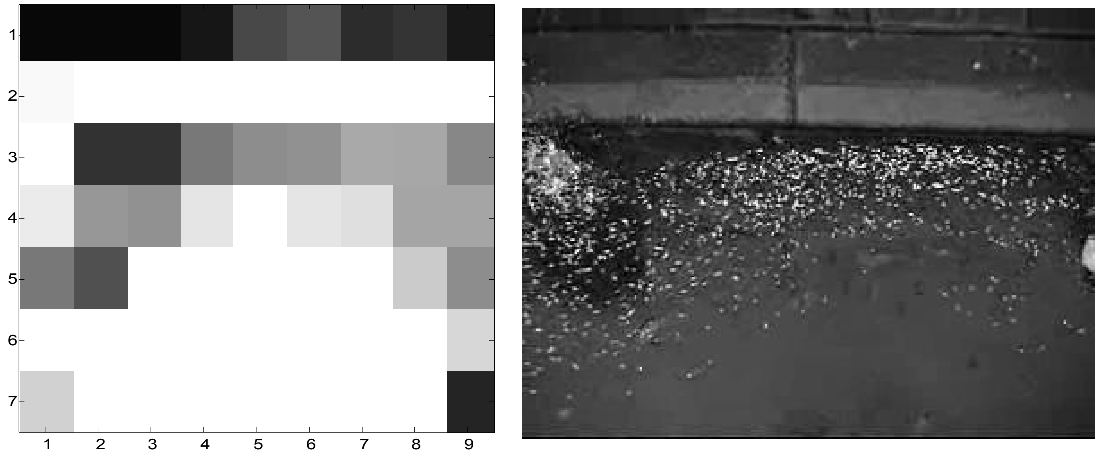


Figure …: Concentrations of traser particles

It can be seen that the backround of the image is sometimes bright and some bright parts with no particles at all are considered to be with large number of particles. To avoid this the part that is the stream bed shoud be found:

```
for i=1:26


I3 =
imread(['C:\MATLAB71\work\urban_discharge\kisa\konf_28\konf_009_000',nu
m2str(i),'.jpg']);
I4 =
imread(['C:\MATLAB71\work\urban_discharge\kisa\konf_28\konf_009_000',nu
m2str(i+1),'.jpg']);


I3=rgb2gray(I3);
I4=rgb2gray(I4);

I{i}=I3-I4;

end


I1=I{1};
for j=2:i
    I1=I1+I{j};
```

```
end
imshow(I1)

level = graythresh(I1);
bw = im2bw(I1,level);
figure
 imshow(bw)

 int_resolution=32;
 bw_out=zeros(min(size(bw)),max(size(bw)));

tresh=80;

 for
kk=1:int_resolution:int_resolution*floor(min(size(bw))/int_resolution)
  for
rr=1:int_resolution:int_resolution*floor(max(size(bw))/int_resolution)
      A=sum(sum(bw(kk:kk+int_resolution-1,rr:rr+int_resolution-1)));
      if A>tresh
      bw_out(kk:kk+int_resolution-1,rr:rr+int_resolution-
1)=ones(int_resolution,int_resolution);
      end
  end
 end

 figure
 imshow(bw_out)
```

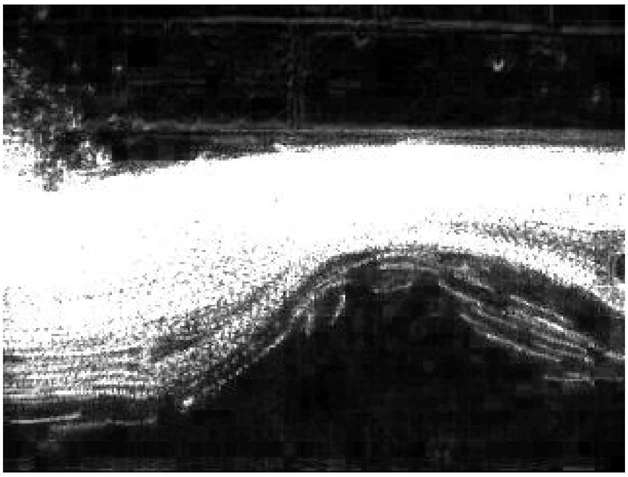


Figure …: Streams of particles

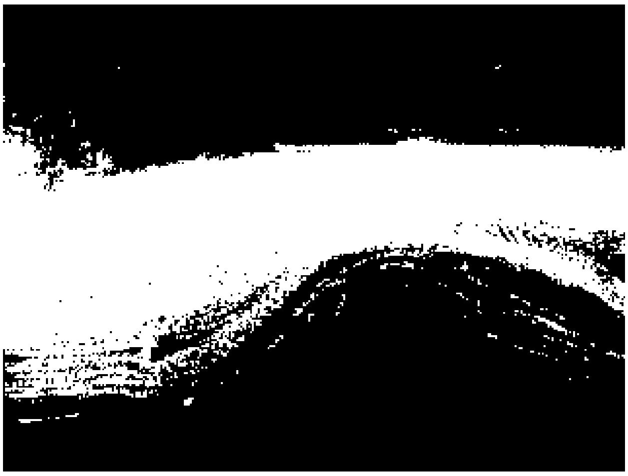


Figure …: Binary image

Figure …: flow bed in 32x32 pixels resolution

```
a=floor(C.*bw_out);
image(a);
colormap(bone);
```

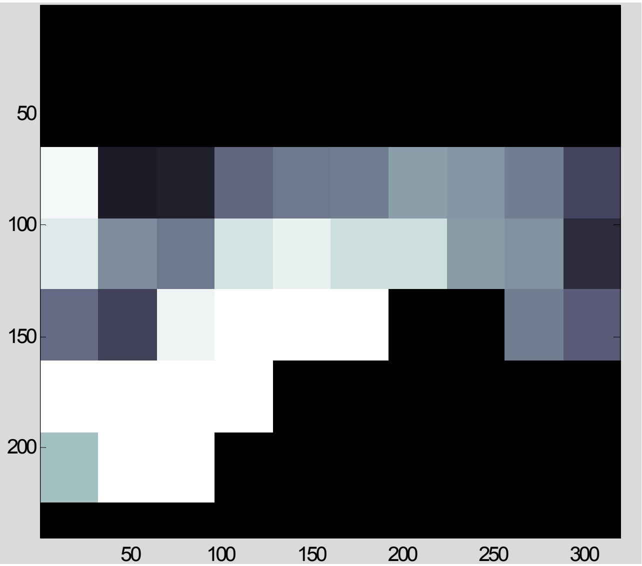


Figure …: concentration of trasers in river bed

Figure …: seeding the stream

For seeding the most gratefull material is paper. Ecologic as it is, it is possible to cut it in the different shapes and sizes. In Figure 3 it can be seen how simple it is to seed the surface of a flow.

Paper seeded should be flat for easier adhesion to the surface. Any shadow might be possible introduction of errors in the captured images.

Grabing frames

Figure 4: two frames of seeded flow

These two frames are captured in time step dt=0.06s. This timestep is highly corelated with the video equipement you own. Also time step is reverce proportional to the velocity that is under investigation. As the time step is smaller the lower velocity can be determined. These two frames are 240x320 RGB pictures.

Setting Interogating Areas

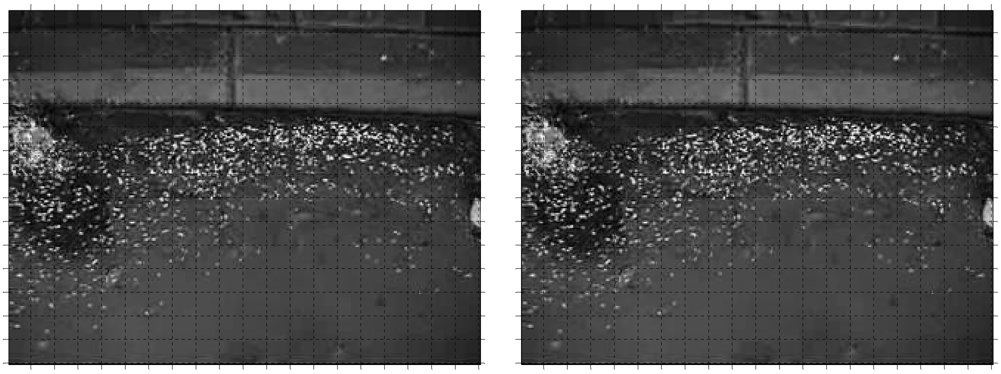


Figure 5: setting grid (interogation areas),16x16 pixels

In Figure 5 the grid is seted to 16x16 pixel size. This is beacouse the 2D FFT algorithm works faster if the input pictures are 2^nx2^n size. every interogation area is compared to the others starting with the neiburing ones using the crosscorelation equations.

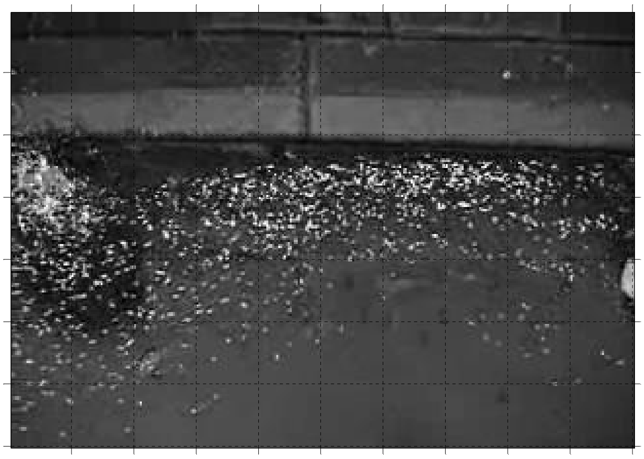


Figure 5: setting grid (interogation areas),32x32 pixels

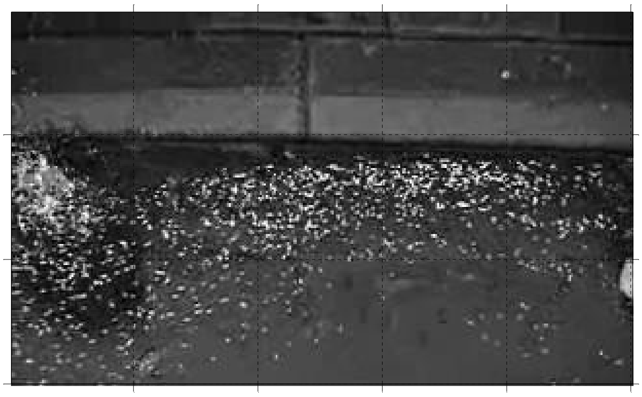


Figure 5: setting grid (interogation areas),64x64 pixels

Calculating corelation

The most used method for finding the corelation betveen two interogation areas is $ifft2(fft2(a)\bullet fft2(conj(b)))$ method, where a and b are the interogation area pictures. The method is presented on folowing Figures:

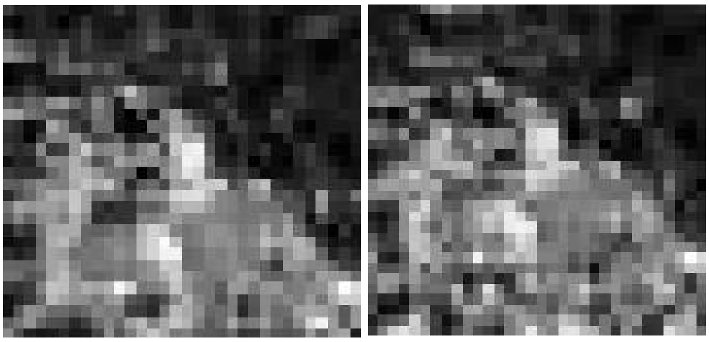


Figure …: two frames of interogation areas 32x32 pixels

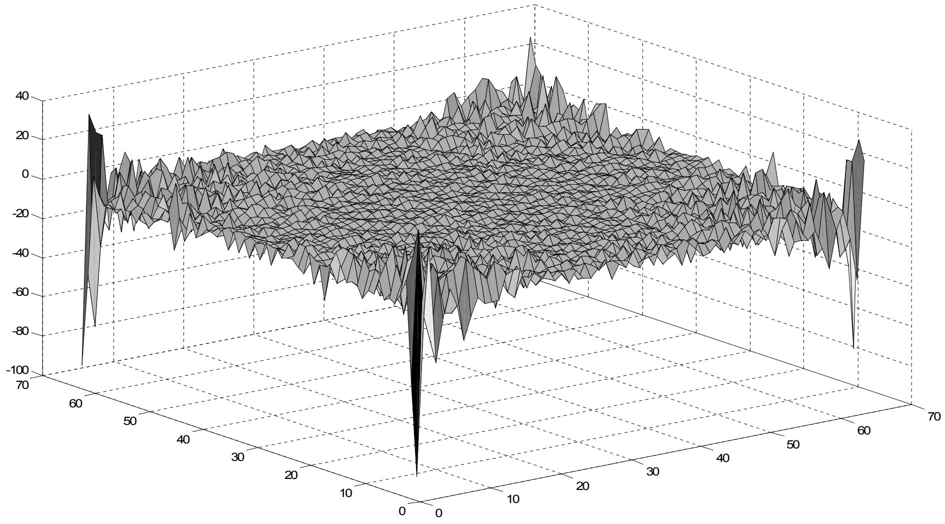


Figure …: output of $fft2(a)$

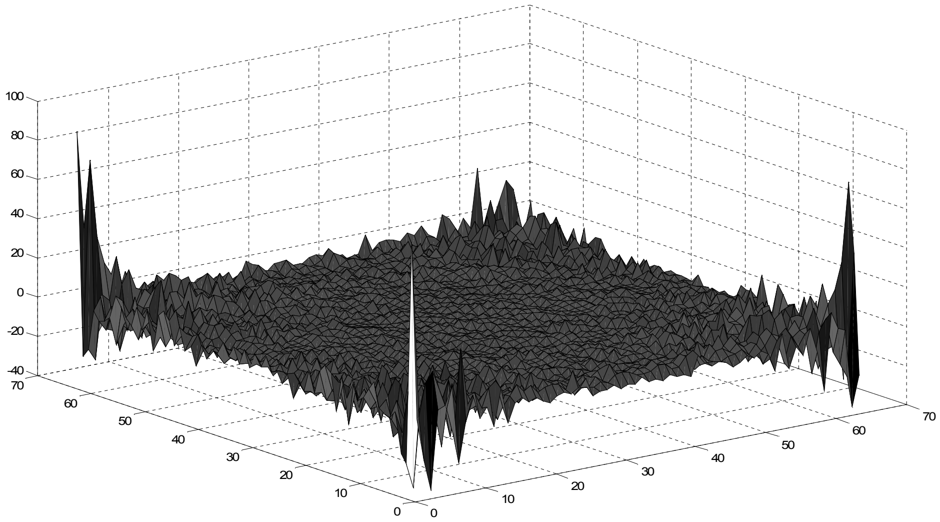


Figure …: output of $fft2(conj(b))$

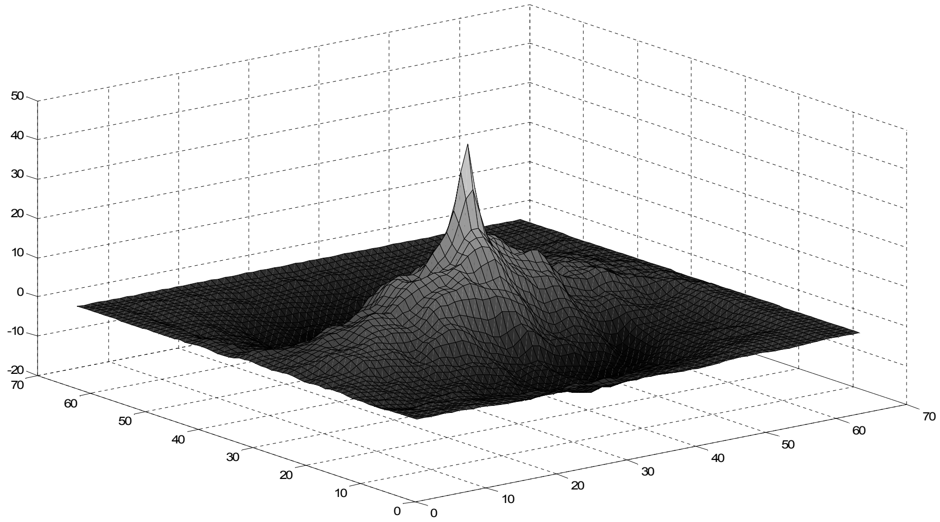
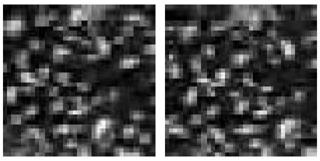


Figure …: output of $ifft2(fft2(a) \bullet ff2t(conj(b)))$

Peak in this output shows how much these two interogation areas mache. Finding the best maching means finding the displacement of the interigation area in the space during time dt.

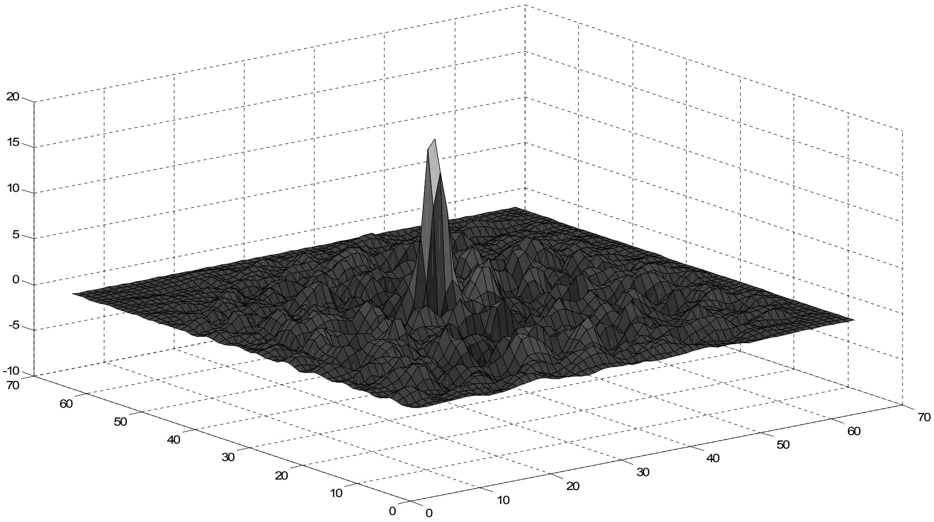
Second way of finding the corelation between two interogation areas is the 2D crosscorelation funtion:

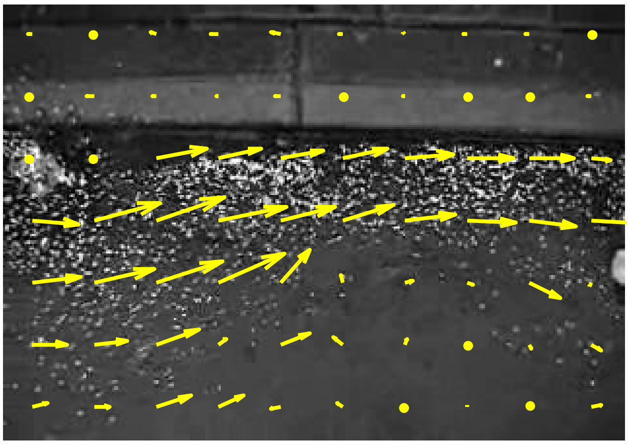
Comparation of two methods of finding crosscorelation



First crosscorelation method is:

$$ifft(fft(a) \bullet fft(conj(b)))$$

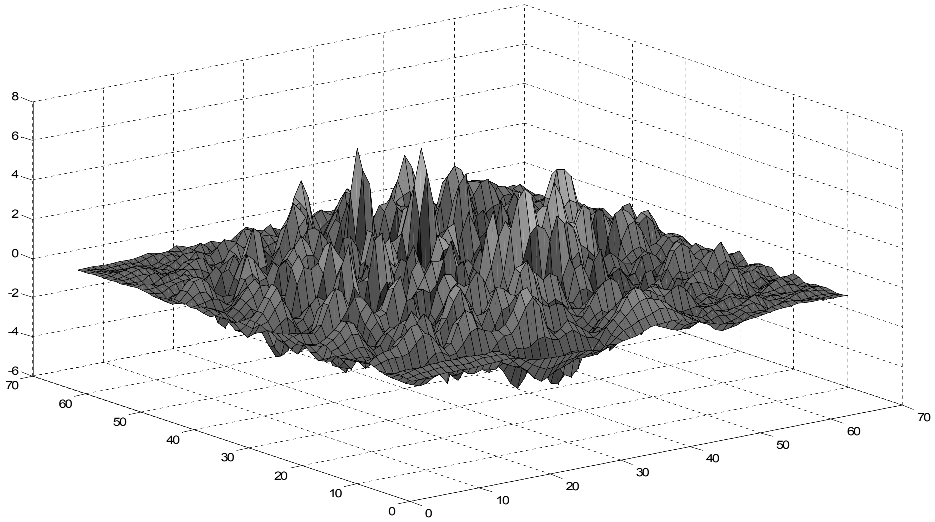


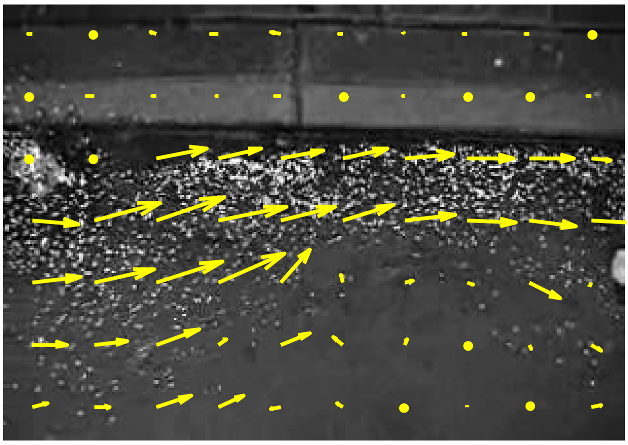


,and the second one:

$$c(x_i, y_i) = \sum_{m=1}^{size(b)-1} \sum_{n=1}^{size(a)-1} A(i,j) \bullet conj(B(m+i, n+j))$$

where: $0 \le i < size(a) + size(b) - 1$ and $0 \le j < size(a) + size(b) - 1$





It can be noticed that the although it is harder to derive peak in crosscorelation matrix, the result is the same.

Output image

Finaly matrix of all displacements can be formed and it can be shown on the image as output image.

The matrix of crosscorelation coeficients is than used for finding the best maching of interogation areas. The best mathes are than connected with the vectors that represent the velocity vectors.

Output image is usualy presented with arrows representing the surface velocity field, like in Figure …

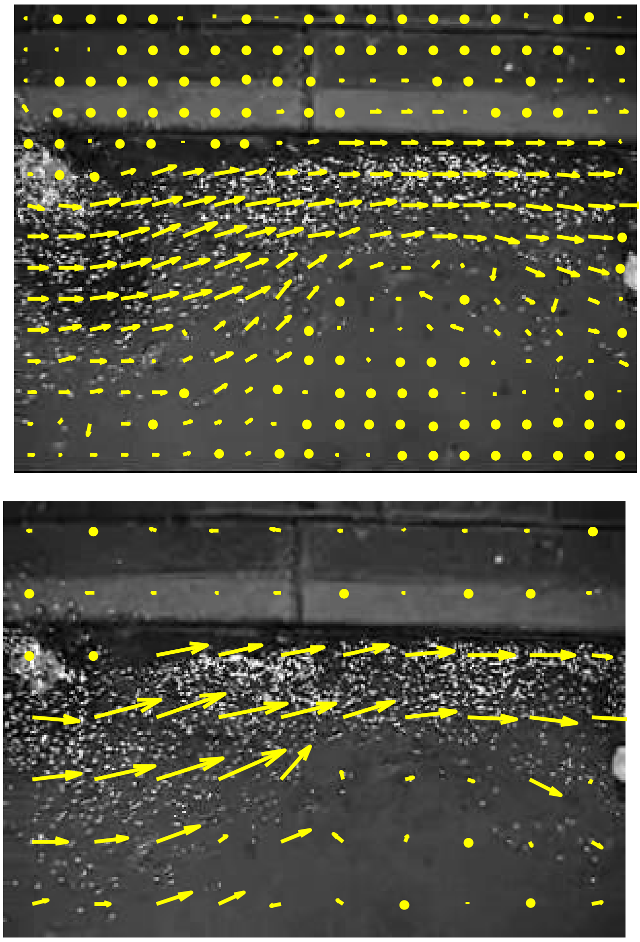
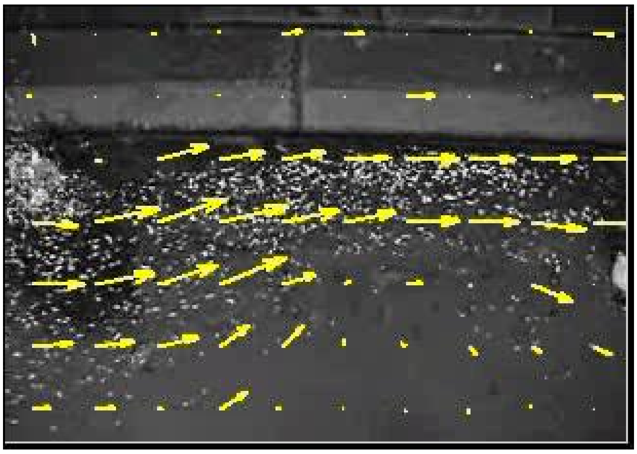
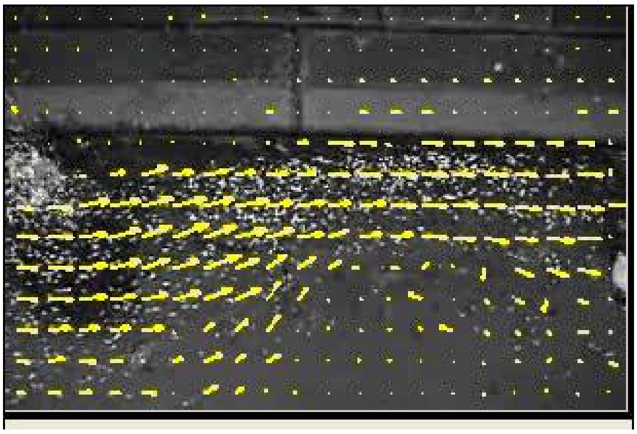


Figure 6: Output of a LSPIV

It can be seen that there are a lot of errorneus vectors, especialy out of the flow stream. This is a product of a picture noise originated from the fact that the camera was held in the arm or from the fact that the raindrops were falling during the rain event. This can be overcome using more frames.



Video clip: 32x32 pixels interogation area



Video clip: 16x16 pixels interogation area

Filtering the velocity data

Next step is to filter the velocity data from the several frames. In figure … is presented filtered data using the mean mean velocities that are larger than some trash value.

```
x=[];
y=[];

u=[];
v=[];
```

```
for ii=1:26
res1=res{ii};
x=[x,res1(:,1)];
y=[y,res1(:,2)];
u=[u,res1(:,3)];
v=[v,res1(:,4)];
end

u(:,9)=u(:,8);
v(:,9)=v(:,8);

% windowSize = 8;
% for kk=1:260
%
% u_filter(kk,:)=filter(ones(1,windowSize)/windowSize,1,u(kk,:));
% v_filter(kk,:)=filter(ones(1,windowSize)/windowSize,1,v(kk,:));
%
% end


% plot(u(1,:))
% hold on
% plot(u_filter(1,:),'r')


srednja_u=(mean(u'))';
srednja_v=(mean(v'))';


trash=0.5;
brzina_trash=find(sqrt(srednja_u.^2+srednja_v.^2)<trash);

srednja_u(brzina_trash)=0;
srednja_v(brzina_trash)=0;


rezultat_filtriranja=[res1(:,1:2),srednja_u,srednja_v,res1(:,5)];


figure; imshow(I3);
   hold on
quiverm(rezultat_filtriranja,'y','LineWidth',2);
```

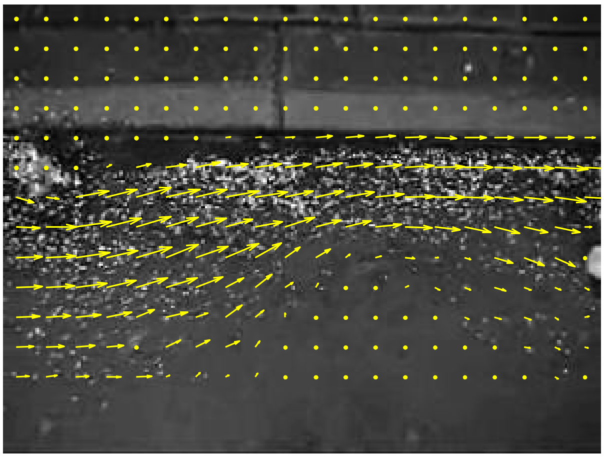


Figure …: Filtrated data

Discharge calculation

This velocity field can be integrated and the discharge can be stimated. This discharge canbe further used in sewage analisys, assesment of discharge coeficients of a catchment or something else…

```
profili_intenziteti=[];
profili_pravci=[];
profili_u=[];
profili_v=[];

for hh=8:16:312

a=find(rezultat_filtriranja(:,1)==hh);

profili_u=[profili_u,rezultat_filtriranja(a,3)];
profili_v=[profili_v,rezultat_filtriranja(a,4)];

profili_intenziteti=[profili_intenziteti,sqrt(rezultat_filtriranja(a,3)
.^2+rezultat_filtriranja(a,4).^2)];

profili_pravci=[profili_pravci,rad2deg(atan(rezultat_filtriranja(a,4)./
rezultat_filtriranja(a,3)))];

end


srednja_prof_brzina=mean(profili_u)';
```

```
protok=sum(profili_u*32)';%[brzina x pixela x dubina]


protok_mean=mean(protok)
protok_std=std(protok)
```