Subject: LOCAL THRESHOLD  / LOCAL MAXIMA / TEMPLATE MATCHING
Posted by stoeckli on Wed, 26 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Hi everybody !

(LOCAL THRESHOLD  / LOCAL MAXIMA / TEMPLATE MATCHING)

In my application, a novel 2-D measurement technique, with which I can visualize droplets of an Diesel injection spray in a running engine, I use an ICCD camera and a pulsed Nd:YAG Lasers as well
as a quartz lightwave guide to lead the laser beam to the combustion chamber. The idea is to compute droplet size and droplet volume density distributions (both contour- and histogram plots) of the investigation area and to correlate these data with NOx and soot emissions as well as fuel consumption.

Due to the fact that the droplet size is between approx. 1 and 80 um and that the illumination cone

is not symmetrically the reflection and refraction (2. order) areas on the droplets are complicated (e.g. egg-shaped with border effects) and of different sizes. Small droplets lead to small reflection/refraction areas and appear much fainter (lower intensity peaks) on the image. Therefore
I can't use a general threshold to distinguish between background/noise and these r/r areas.

>>>  Does anyone know of a function / procedure that generates a LOCAL THRESHOLD around local maxima?

>>>  Does anyone know of a function to find LOCAL MAXIMA in a 2-D image? (Something like nr_powell resp. powell but 2 dimensional)

So far I'm using  a global threshold. After several repetitive erode/dilate operation to get rid of single pixel "noise" and of area "bridges", I'm counting the r/r areas, determine their sizes and compute the real droplet diameter.

This of course is not very refined. A much better approach would be to low-pass filter the original image and to use a TEMPLATE MATCHING function.

>>>  Has anyone ever programmed such a TEMPLATE MATCHING function? The function I'm looking for is related with the convolution funct=
ion, but with another mathematical operation:

f = Sum[0..m] (sqr(ztemplate[i+m] - zimage[j+m]))     (written for 1 dimension)

A perfect match results in f = 0 at the actual template position.

Thanx in advance

Regards,

Martin

P.S.: I'm using IDL 4.0 on a P90 PC

**********

Martin Stoeckli
Swiss Federal Institute of Technology
I.C. Engines and Combustion Laboratory
ETH Zurich - IET / LVV Tel: +41 1 632 4981
Sonneggstrasse 3 Fax: +41 1 632 1177
CH-8092 Zurich e-mail: stoeckli@lvv.iet.mavt.ethz.ch
Switzerland

**********