

ZAPDC. Ćwiczenie 1. Aliasing 2D

Lev Sergeyev

2019.03.06

1 Przebieg ćwiczenia

Za pomocą środowiska Matlab wygenerowałem animację składającą się z 60 klatek która przedstawia obracające się śmigło (Rys 1). /par do generacji obrazu animacji służą następujące zmienne

- X_r i Y_r : rozdzielczość obrazu
- F_N : ilość klatek
- propeller: ilość ramion śmigła

Rysunek 1: Animacja: obracające śmigło obraz oryginalny (rzeczywisty)

Następnie przetworzyłem otrzymaną animację w taki sposób, aby symulować odczytywanie obrazu sensorem kamery cyfrowej: każda klatka dzieli się na VS odcinków pionowych w które są zapisywane odpowiednie odcinki z opóźnieniem. W taki sposób otrzymałem animację (Rys. 2) o częstotliwości $f = VS/F_N$, np jeżeli częstotliwość animacji pierwotnej jest 60 Hz, to częstotliwość po przetwarzaniu jest $60/VS$ Hz.

Rysunek 2: Animacja: obraz pobrany z matrycy

Wygenerowane animacje (F.gif i A.gif) są z uwzględnieniem proporcji czasowej, w taki sposób aby chwila trwania jednej klatki w A.gif odpowiadała VS klatkom w F.gif

2 Wnioski

Takie zniekształcenia są skutkiem opóźnienia odczytu poszczególnych pól sensoru. Aby uniknąć takie zniekształcenia należało by odczytywać wszystkie piksele na raz, podnieść częstotliwość odczytu, albo wprowadzić post processing (np całkowanie, co swoją drogą też może wprowadzić pewne zniekształcenia w postaci śladów na obrazie statycznym). W niektórych sensorach i algorytmach kodowania sygnał również stosowany był odczyt obrazu co drugi wiersz, co również powodowało powstanie pasków poziomy w nagraniach.

3 Kod źródłowy

```
1  close all;
2  clear;
3
4  propeller = 3;
5  %Resolution:
6  Xr = 320;
7  Yr = 240;
8  %Frames:
9  F_N = 60;
10 %VeScans_per_frame:
11 VS = 6;
12
13 for i = 1 :(F_N)
14     m = i - 31;
15     theta = 0:0.01:2*pi;
16     rho = sin(propeller*theta + m*pi/10);
17
18     [x,y] = pol2cart(theta, rho);
19     f = figure('visible', 'off', 'Position', [0 0 Xr Yr]);
20     plot(x,y,'k')
21     xlim([-1 1])
22     ylim([-1 1])
23
24     F(i) = getframe;
25 end
26
27 A = F;
28 size_v = size(A(1).cdata(:,1,1));
29 line_h = size_v(1)/VS;
30 for i = 1 : F_N*VS
31     for vl = 1:VS
32         line_begin = round(1 + line_h * (vl-1));
33         line_end = round(line_h * vl);
34         A(i).cdata(line_begin:line_end, :, :) = F(1+mod(i*vl, F_N)).
            cdata(line_begin:line_end, :, :);
35     end
36 end
37
38 %implay(A)
39 %implay(F)
40 movie2gif(F, 'F.gif', 'LoopCount', Inf, 'DelayTime', 1/F_N)
41 movie2gif(A, 'A.gif', 'LoopCount', Inf, 'DelayTime', VS/F_N)
```