

DeGrainMedian 0.8.2

Scritta da buzzqw il 26-09-2007

Il filtro DeGrainMedian e' da usare principalmente per rimuovere il grain, ma puo' essere usato anche come denoiser generalizzato. Usa qualche metodo spazio-temporale mediano.

Questa è la traduzione del manuale ufficiale.

DeGrainMedian 0.82

Traduzione del Manuale Ufficiale

- SCRITTA DA : *...:Buzzqw:...*

- COLLABORAZIONE, SUPPORTO E REVISIONE : *...:DivXmania Staff:...*

Filtro Spazio-Temporale Mediano per la rimozione del grain

Plugin per Avisynth 2.5, [HomePage](#)

Version 0.8.2 [Download](#)

Copyright ©2004-2006 Alexander G. Balakhnin (aka Fizick).

<http://avisynth.org.ru>

Il filtro DeGrainMedian e' da usare principalmente per rimuovere il grain, ma puo' essere usato anche come denoiser generalizzato.

Usa qualche metodo spazio-temporale mediano.

Sintassi

DeGrainMedian(clip, int "**limitY**", int "**limitUV**", int "**mode**", bool "**interlaced**", bool "**norow**")

Parametri del Filtro

Il primo parametro e' sempre la clip di input o l'ultimo default.

limitY (da 0 a 255, default=4) - indica il limite per la correzione del luma per pixel

limitUV (da 0 a 255, default=6) - indica il limite per la correzione del chroma per pixel

mode (da 0 a 5, default=1) - indica la modalita' operativa (piu' e' alto, piu' e' leggero il filtro)

interlaced (true o false, default=false) - tratta la fonte come interlacciata o meno

norow (true o false, default=false) - non usa la stessa riga nel filtro spaziale

Uso per la rimozione del grain e noise

Carica il plugin ed aggiusta i parametri

Il **limitY** assume valori attorno 2-8. Questi valori sono un buon compromesso tra il denoise e il blur.

Sembra che valori piu' alti del **limitUV** per il chroma (4-15) possano essere comunque sicure.

Se l'immagine ti sembra troppo appiattita prova ad alzare il valore di **mode**!

Piu' e' alto il valore di **mode**, piu' e' sicuro che valori alti non facciano danni

Sembra che **mode=1** o **mode=2** siano un buon compromesso, ma ovviamente dipende dalla sorgente.

Con **mode=5** i cambiamenti apportati sono pressoché invisibili, ma alcuni pixel "caldi" vengono rimossi.

L'opzione **norow=true** e' utile con video che hanno righe orizzontali, come acquisizioni analogiche da TV o VHS

Semplici esempi di script

Filtro leggero, ma utile, per la rimozione di pixel caldi:

CODE

```
Avisource("dvd-input.avi")
loadplugin("DeGrainMedian.dll")
DeGrainMedian(limitY=5,limitUV=5,mode=3)
```

Filtro a due passi per la rimozione piena del rumore:

CODE

```
Avisource("input.avi")
loadplugin("DeGrainMedian.dll")
DeGrainMedian(limitY=2,limitUV=3,mode=1)
DeGrainMedian(limitY=2,limitUV=3,mode=1)
```

Applicazione forte del filtro per una grossa rimozione del grain:

CODE

```
Avisource("analog-input.avi")
loadplugin("DeGrainMedian.dll")
DeGrainMedian(limitY=5,limitUV=7,mode=0)
```

Filtraggio consigliato per rimuovere le strisce orizzontali:

CODE

```
Avisource("analog-input.avi")
loadplugin("DeGrainMedian.dll")
DeGrainMedian(limitY=5,limitUV=7,mode=1, norow=true)
```

Dettagli tecnici

Questo plugin e' principalmente basato su due idee, usate nelle due fasi del processo.

Prima fase - Il Rilevamento

La prima idea e' presa dal filtro *STMedianFilter* di Tom Barry

Io ho anche usato parte della suo codice estremamente ottimizzato e messo qui parte del suo manuale (modificato un poco):

QUOTE

STMedianFilter e' un filtro mediano (leggermente compensato dal moto) spazio/temporale. E piuttosto leggero nelle operazioni, usando solo i pixel adiacenti nella spazio e tempo, cosi' controlla le 26 (*NDT: immagina un cubo di 3x3, con il pixel da analizzare al centro*) locazioni intorno per filtrare ogni parte. Filtra anche il luma e chroma...

...Un semplice filtro mediano e' giusto una salto di gradino dove il valore e' impostato affinche' non superi il grado superiore ed inferiore dei vicini.

...(probabilmente per essere piu' corretti un mediano e' il membro medio di una serie ordinata di valori - Fizick).

Per esempio se tu hai 3 pixel in fila che hanno valori $\langle 5,8,7 \rangle$, tu puoi impostare il valore centrale per non essere fuori dai valori 5 o 7. Quindi un valore possibile potrebbe essere 7.

Ora immafine di avere un piccolo video 3x3, come la superficie di un cubo di Rubik. Immagina che il frame precedente sia i fondo del cubo, il frame corrente e' lo strato del mezzo e lo strato superiore e' prossimo frame.

Quindi il pixel corrente sara' proprio il pixel centrale del cubo di Rubik a ci saranno 13 modi per cui puoi tracciare una linea attraverso lui ed un paio di pixel attorno piu' vicini.

Quello che io (e io Fizick) facciamo e' comparare ogni coppia di questi pixel prossimi per vedere quale paio e' piu' in accordo sui valori. I (ed io Fizick) usiamo quella coppia per impostare il valore del pixel centrale

(Fine della citazione da Tom Barry).

E' molto interessante che il plugin *RemoveGrain* di Kassandro, nella modalita *mode=9*, e' molto simile a *STMediaFilter*, ma solo sul piano spaziale e non temporale. *Removegrain* poi cambia il valore del pixel basandosi su un minmax dei valori dei pixel vicini. Ma attenzione, non puo' rimuovere valori medi di grain dalle mie fonti video molto rovinare.

RemoveGrain e' considerato da Kassandro come un prefiltro per *RemoveDirt*. Nel manuale viene detto:

QUOTE

Se il grain e' troppo forte, *RemoveGrain* puo' solo rimuovere parzialmente o per nulla il grain

. (Fine dalla citazione di Kassandro)

Molto probabilmente questa mancanza e' dovuta alla mancanza di informazioni temporali dai frames circostanti.

Sin dalla versione 0.2 ho aggiunto a *DeGrainMediam* anche alcune modalita' operative sicure 1-4 (e piu' tardi 5) (codice preso da *RemoveGrain*).

Ora consideriamo sempre il solito cubo 3x3x3.

I nuovi valori proposti per il pixel (**newp**) sono dunque presi dai valori presi dalle coppie di pixel vicini (*bound1* e *bound2*).

Le modalita' mode=1 e mode=5 usano un criterio piu' sicuro per pesare la coppia ottimale dei pixel. Tale valore tiene conto delle differenze tra i nuovi valori del pixel centrale ed i valori vecchi (**oldp**)

- **Mode=0** di DeGrainMedian (simile al mode=9 di RemoveGrain) usa $\text{weight} = |\text{bound1} - \text{bound2}|$. E' il piu' forte
- **Mode=1** di DeGrainMedian (piu' forte di mode=8 di RemoveGrain) usa $\text{weight} = |\text{oldp} - \text{newp}| + 4 * |\text{bound1} - \text{bound2}|$
- **Mode=2** di DeGrainMedian (simile a mode=8 di RemoveGrain) usa $\text{weight} = |\text{oldp} - \text{newp}| + 2 * |\text{bound1} - \text{bound2}|$
- **Mode=3** di DeGrainMedian (simile a mode=7 di RemoveGrain) usa $\text{weight} = |\text{oldp} - \text{newp}| + |\text{bound1} - \text{bound2}|$
- **Mode=4** di DeGrainMedian (simile a mode=6 di RemoveGrain) usa $\text{weight} = 2 * |\text{oldp} - \text{newp}| + |\text{bound1} - \text{bound2}|$
- **Mode=5** di DeGrainMedian (simile a mode=5 di RemoveGrain) usa $\text{weight} = |\text{oldp} - \text{newp}|$. E' il piu' debole

L'opzione **norow** esclude le copie di pixel orizzontali per lo stesso frame. Quindi il filtro esamina solo 12 coppie possibili.

Seconda fase - Il Filtraggio

STMedianFilter nella seconda fase non filtra il pixel se questo lo cambia piu' di un certo valore di soglia. (Piu' correttamente prima fa un rilevamento temporale e di soglia, poi seguito da un rilevamento spaziale e di soglia)

La maggior parte dei denoiser non filtra pixel con grossi rumori (grandi differenze di luma). Ma questi singoli pixel sono estremamente visibile se sono presenti in frami piatti ed omogenei e gruppi di questi pixel creano artefatti sui bordi veramente brutti.

Oltretutto un filtro di riduzione del rumore solamente temporale produrrebbe effetti ghost anche per soglie non troppo elevate.

DeGrainMedian in questa fase usa un differente metodo di approccio per processare il pixel.

Qui io usa la seconda idea, presa dal filtro Dust di Steady. Vediamo parte della documentazione del filtro Dust:

"Limit=5

QUOTE

Imposta la forza del filtraggio temporale. (Quanto puo' cambiare il pixel originale)

(Fine della citazione da Steady).

Quindi se il pixel filtrato cambiasse piu' della soglia (**limit**) non sarebbe ripristinato al valore originario MA cambiato al valore del limite.

DeGrainMedian filtra tutti i pixel, ma con forze piu' limitata.

Quindi tutto il grain e strikes sono (completamente o parzialmente) ripuliti.

Il filtro non dovrebbe produrre artefatti ai bordi immagine. Anche il ghost e' minimo dato che automaticamente passa alla pulizia spaziale appena i valori delle coppie si avvicinano.

Puoi scegliere uno dei sei modi operativi, dal forte (ma "lavato") al debole (ma "definito").

In definitiva DeGrainMedian ha la velocita' di STMedianFilter e la forza di Dust (limitato dai valori :-)! E non e' uno scherzo :-).

Puo' veramente migliorare la compressione del video.

Allora cosa c'e' di brutto ?

1. Un po' di blur (ma vorresti rimuovere il rumore senza il blur ? :-) .
2. Non puo' produrre superfici perfettamente appiattite, un po' di rumore rimarra' sempre (ti piacciono i filmato super piatti e blocchettosi ?)
3. La compensazione di moto e' limitato ad 1 pixel (ovviamente questo non e' Dust, ma comunque puoi sempre usare un filtro moto compensato esterno ?).

Seramente, questo non e' un super filtro ed e' ancora sperimentale, ma per i miei film granulosi ha dato degli ottimi risultati.

Inizialmente lo avevo sviluppato come prefiltro (a Vaguedenoiser ed altri) ma mi sembra ora che possa essere usato da solo...

Visto che probabilmente puo' aiutare qualcun altro ho deciso di rilasciarlo.

Alcune note.

Con limitY=255, DeGrainMedian deve avere lo stesso output che STMedianFilter con thresholds=255 (e con pure gli stessi artefatti), ma ad adesso non e' vero... probabilmente dovuto a qualche buco... o buchi :-)

Correzione: adesso e' vero. Era un buco di STMedianFilter. L'ho messo a posto nella versione 1.0.3.

Uso combinato

Deve essere usato prima di altri filtri di denoise.

Qualche volta io uso due istanze di DeGrainMedian con limiti piu' bassi per una migliore rimozione del rumore.

I migliori risultati per la riparazione dei film con AviSynth possono essere raggiunte con il mio plugin di

compensazione globale di moro *DePan*.

Abilita' e limitazione delle versione corrente

- Funziona solo nei campi di colore YV12 e YUY2
- Necessita di un processore P-III, Athlon o migliore. Necessita delle istruzioni SSEMMX.
- Non ha un rilevatore di cambi di scena ma non mi e' mai capitato di vedere effetti ghost. Ricordo che il filtro passa alle coppie spaziali
- La prima e l'ultima line (coppia di linee in caso di video interlacciati) non sono processati
- Il primo e gli ultimi 8 (YV12) o 4 (YUY2) pixel di ogni riga sono ora filtrati con un metodo piu' semplice.
- Il filtraggio del chroma nella modalita YUY2 e' anche solo nelle stesse colonne
- Il primo e l'ultimo frame non sono processati
- Testato con Avisynth 2.5.5
- E' veramente un filtro veloce

Informazioni aggiuntive

Il thread ufficiale e' sul forum di Doom9

<http://forum.doom9.org/showthread.php?s=&t...&threadid=80834>

Compilazione

Per compilare i uso il compilatore gratuito MS VC++ Toolkit 2003 con la piattaforma MS SDK.

Il comando makefile viene usato con l'opzione

nmake

(nota: copia il file nmake.exe e cvtres.exe dalla Platform SDK Bin\win64 alla Bin).

Licenza

Questo programma è un software libero, è possibile ridistribuirlo e/o modificarlo in conformità alle disposizioni della Licenza PubblicabGenerale GNU, come viene pubblicata dalla Free Software Foundation; versione 2 della licenza o (a scelta) tutte le versioni successive.

Questo programma è distribuito nella speranza che possa essere utile, ma SENZA ALCUNA GARANZIA, e senza garanzia implicita di COMMERCIALIZZAZIONE o di ADATTAMENTO A UN OGGETTO PARTICOLARE. Per maggiori dettagli vedi la Licenza Pubblica Generale GNU .

Un esemplare della Licenza Pubblica Generale GNU deve essere fornito con questo programma; in caso contrario scrivere a Free Software Foundation Inc., 675 Mass Ave, Cambridge, MA 02139 USA.

Se il mio plugin ti piace, ti progo di considerare una donazione! (NDT: io l'ho fatto, e prossimamente lo rifaro')

Version changes:

- * Version 0.1, August 13, 2004 - first public (beta!).
- * Version 0.1.1, August 18, 2004 - Bug fixed with spatial part of YV12 (distance was 2 pixels)
- * Version 0.2, August 21, 2004 - Added more safe modes=1-4 , doc updated
- * Version 0.3, August 28, 2004 - Added interlaced mode, code reorganization
- * Version 0.4, September 10, 2004 - Fixed possible bug with different frame pitches
- * Version 0.5, March 23, 2005 - Use more correct cache settings. Thank to Turyst04 for stability problem report.
- * Version 0.6, April 2, 2005 - fixed some bugs with YUY2. Thanks to AVIL for report.
- * Version 0.7, April 21, 2005 - fixed bug with mode=2 for YV12 interlaced; first frame pitch fixed
- * Version 0.7.0 July 27, 2005 - reformated documentation only
- * Version 0.8 October 9, 2005 - changed frame cache range to 3;
 - added option norow to disable using same row (horizontal) spatial pair;
 - fixed documentation error about mode=1;
 - renamed old mode 4 to 5, 3 to 4, 2 to 3, added new mode 2.
- * Version 0.8.1 July 28, 2006 - Fixed a bug for mode>0, interlaced=true, norow=false, YV12. Thanks to squid_80 for report.
- * Version 0.8.2 October 7, 2006 - Fixed a green right border bug for non-mod 8 width. Thanks to akapuma for report.

Buzzqw