

Texturkomprimierung

Philipp Klaus Krause

6. November 2007

1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

3 Vergleich

4 Bibliographie

1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

3 Vergleich

4 Bibliographie

Motivation

- Begrenzter Graphikspeicher
- Speicherbandbreite als Flaschenhals
- Bandbreite ist teuer, kostet Energie
- Hauptanteil des Speicher- und Bandbreitenbedarfs entfällt auf Textur(zugriffe)

Ausweg: Texturkomprimierung

- Geringerer Bandbreiten- und Speicherbedarf um einen gegebene Bildqualität zu erreichen
- Höhere Bildqualität ohne erhöhten Bandbreiten- und Speicherbedarf

Anforderungen an Texturkomprimierung

Schneller Zugriff auf zufällige Texel ist nötig. Deshalb:

- Feste Komprimierungsrate
- Wenige Indirektionen
- Erhalt der Lokalität
- Unkomplizierter, gut in Hardware zu implementierender Dekomprimierungsalgorithmus

1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

3 Vergleich

4 Bibliographie

Bedeutende Verfahren

- Indizierte Farben
- Vektorquantisierung
- S3TC, (VTC)
- FXT1
- 3Dc
- latc, rgtc
- PVRTC
- ETC/iPACKMAN, (PACKMAN)
- ETC2
- ATC

1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

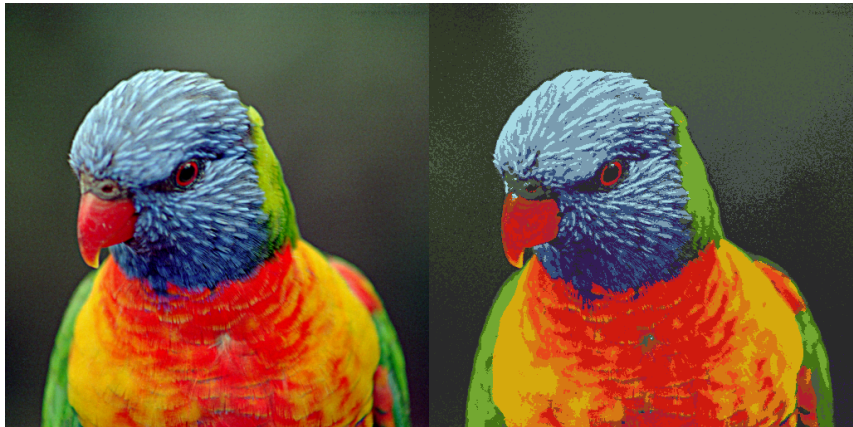
3 Vergleich

4 Bibliographie

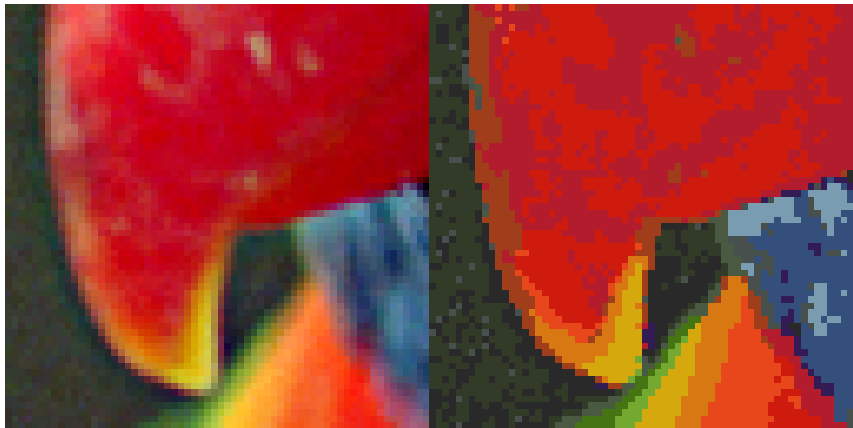
Indizierte Farben

- Ältestes Verfahren
- Auch allgemein bei der Bildkomprimierung genutzt
- Sonderfall der Vektorquantisierung
- Pro Texel ein Index in eine texturspezifische Farbtabelle

Indizierte Farben



Indizierte Farben



1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

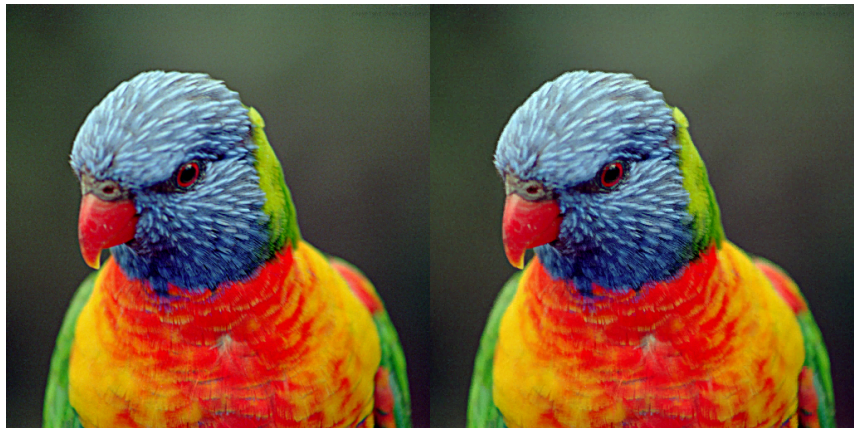
3 Vergleich

4 Bibliographie

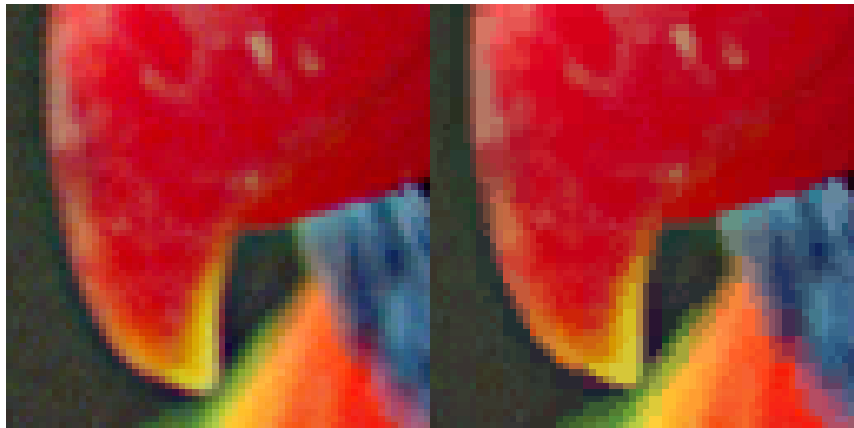
- Meistgenutztes Verfahren
- In DirectX aufgenommen
- Patentiert
- 5 Verfahren, die sich bei Transparenz unterscheiden.

- Je 4x4 Texel werden in einem 64-Bit Datenblock gespeichert
- Jeder Block enthält zwei 16-Bit Farbwerte (RGB565)
- Jeder Block enthält einen 2-Bit Wert pro Texel
- Dieser 2-Bit Wert wird verwendet um durch Interpolieren zwischen den Farbwerten die Farbe des Texels zu erhalten
- Die Reihenfolge der 1-Bit Farbwerte kodiert die Opazität des Blocks

DXT1



DXT1



1 Einleitung

2 Verfahren

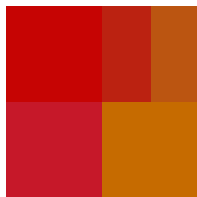
- Indizierte Farben
- S3TC
- ETC/iPACKMAN

3 Vergleich

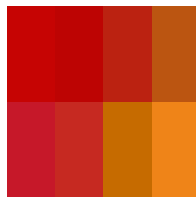
4 Bibliographie

- Menschen können Luminanz- besser als Chrominanzunterschiede erkennen.
- Die Helligkeit einer Grundfarbe wird variiert.

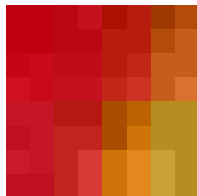
- Jeder 4x4-Block wird in zwei 2x4 oder 4x2-Unterblöcke zerlegt. Jeder Unterblock hat eine Grundfarbe.
- Die beiden Grundfarben sind entweder je als RGB444 gespeichert oder als RGB555 für die erste und eine RGB333 Differenz.
- Pro Unterblock wird eine Liste mit 4 Einträgen gewählt, die angibt, wie stark die Helligkeit der Texel variiert.
- Für jeden Texel wird ein 2-Bit Wert gespeichert, der eine Helligkeitsänderung aus der Tabelle wählt.



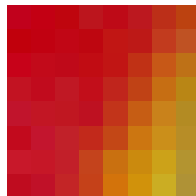
(a) Grundfarben
(direkt)



(b) Grundfarben
(Differenz)

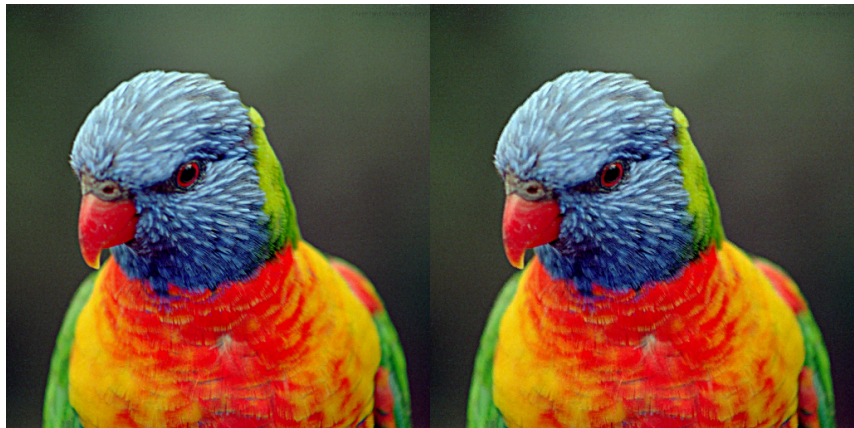


(c) Dekomprimier-
tes Bild

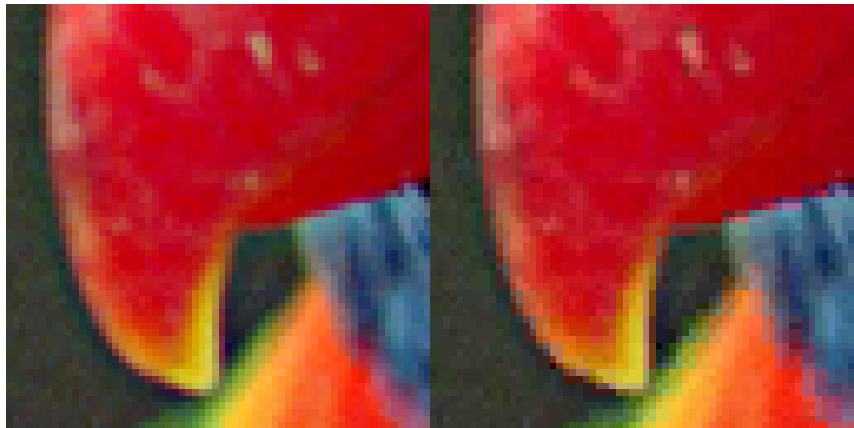


(d) Ursprüngliches
Bild

ETC



ETC



1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

3 Vergleich

4 Bibliographie

Vergleich

Indizierte Farben:

- Gut bei Texturen, in denen wenige Farben vorkommen
- Schlecht bei allem anderen

S3TC:

- Relativ gut bei Übergängen zwischen verschiedenen Farbtönen
- Nur 4 Farben pro 4x4-Block

ETC:

- Gut bei Flächen einheitlichen Farbtons, deren Helligkeit variiert
- Schlecht bei Farbverläufen, insbesondere zwischen Farben gleicher Helligkeit und bei benachbarten unterschiedlichen Farben

Es gibt einen Nachfolger, ETC2, der sich dieser Probleme annimmt.

1 Einleitung

2 Verfahren

- Indizierte Farben
- S3TC
- ETC/iPACKMAN

3 Vergleich

4 Bibliographie

Bibliographie I

- [1] 3dfx FXT1 Texture Compression White Paper.
http://www3.sharkyextreme.com/hardware/articles/99/3dfx_fxt/wp/.
- [2] ATI Radeon X800 3Dc White Paper.
<http://www.ati.com/products/radeonx800/3DcWhitePaper.pdf>.
- [3] Andrew C. Beers, Maneesh Agrawala, and Navin Chaddha.
Rendering from compressed textures.
In SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 373–378, New York, NY, USA, 1996. ACM.

Bibliographie II

- [4] Graham Campbell, Thomas A. DeFanti, Jeff Frederiksen, Stephen A. Joyce, and Lawrence A. Leske.
Two bit/pixel full color encoding.
In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 215–223, New York, NY, USA, 1986. ACM.
- [5] Simon Fenney.
Texture compression using low-frequency signal modulation.
In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 84–91, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

Bibliographie III

- [6] G. Knittel, A. Schilling, A. Kugler, W. Straßer.
Hardware for Superior Texture Performance.
In *Computers & Graphics 20*, pages 475–481, 1996.
- [7] T. Inada and M. D. McCool.
Compressed lossless texture representation and caching.
In *GH '06: Proceedings of the 21st ACM SIGGRAPH/Eurographics symposium on Graphics hardware*, pages 111–120, New York, NY, USA, 2006. ACM.
- [8] Jacob Ström and Martin Pettersson.
ETC2: texture compression using invalid combinations.
In *GH '07: Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 49–54, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

Bibliographie IV

- [9] Aaftab Munshi.
Block-based image compression method and apparatus.
US Patent 20060215914.
- [10] Pat Brown.
EXT_texture_compression_s3tc.
http://opengl.org/registry/specs/EXT/texture_compression_s3tc.txt.
- [11] Pat Brown, Mathias Agopian.
EXT_texture_compression_dxt1.
http://opengl.org/registry/specs/EXT/texture_compression_dxt1.txt.

- [12] Jacob Ström and Tomas Akenine-Möller.
iPACKMAN: high-quality, low-complexity texture compression
for mobile phones.
*In HWWs '05: Proceedings of the ACM
SIGGRAPH/EUROGRAPHICS conference on Graphics
hardware*, pages 63–70, New York, NY, USA, 2005. ACM.
- [13] Jacob Ström and Tomas Akenine-Möller.
Packman: Texture compression for mobile phones.
[http://www.cs.lth.se/home/Tomas_Akenine_Moller/
pubs/packman_sketch.pdf](http://www.cs.lth.se/home/Tomas_Akenine_Moller/pubs/packman_sketch.pdf).

[14] Li-Yi Wei.

Tile-based texture mapping on graphics hardware.

In *HWWS '04: Proceedings of the ACM*

SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, pages 55–63, New York, NY, USA, 2004. ACM.