

Hinweise zur Hausarbeit WS15

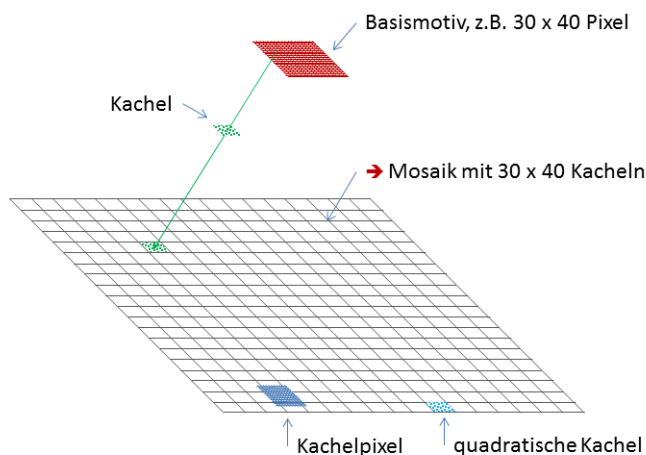
Algorithmen zur Erstellung von Fotomosaiken

Als Erfinder der Fotomosaiken gilt Robert Silvers (<http://www.photomosaic.com/>), der 1995 am MIT einen entsprechenden Algorithmus entwickelte:

<http://www.nbbmuseum.be/de/2014/12/pointillism-in-the-digital-era-robert-silvers-photomosaic.htm> .

Er ließ sich das Verfahren und den Begriff *Photomosaic* patentieren, sodass bis heute nur wenig über die Details bekannt ist.

Hier soll ein Verfahren umgesetzt werden, das möglichst wenig Rechenzeit erfordert und einfach zu implementieren ist. Die Basismotive sollen deshalb nur geringe Auflösung haben, z.B. 30 x 40 Pixel. Jeder dieser Bildpunkte wird durch eine Kachel ersetzt. Somit müssen die Kacheln unbedingt quadratisch sein (z.B. 20 x 20 Pixel), um kein verzerrtes Höhen-/Breitenverhältnis zu erhalten.



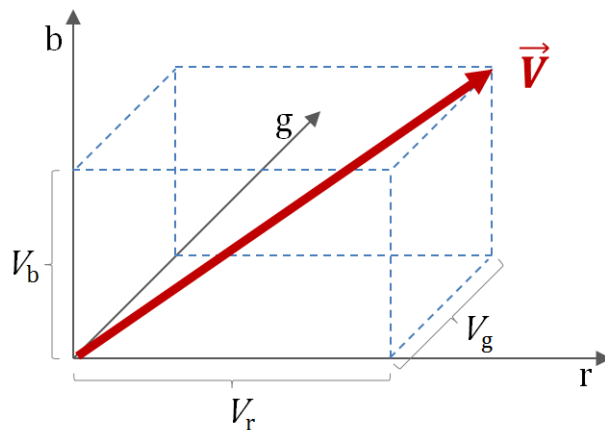
a) Kacheln bester Helligkeit suchen

Ein einfaches Verfahren wird in <http://blog.wolfram.com/2008/05/02/making-photo-mosaics/> beschrieben. Es werden Basismotive und Kacheln mit niedriger Auflösung verwendet. Auch werden die Kacheln nur anhand ihrer mittleren Helligkeit selektiert. Dennoch ist das Ergebnis erstaunlich gut, wenn man ein geeignetes Original und einen dazu passenden Kachelpool verwendet. Schwarzweiße Basismotive eignen sich besonders gut.

- Die Farben aller Bildpunkte seien als RGB-Triple definiert.
- Die Helligkeit eines Pixels ist die Länge des Vektors V mit den Komponenten V_r , V_g und V_b .
- Für alle Pixel des Basismotivs wird die Helligkeit berechnet.
- Da Kacheln aus mehreren Pixeln bestehen, wird zunächst die mittlere Farbe bestimmt. Sie hat die Komponenten: r_{mittel} , g_{mittel} , b_{mittel} . Aus dieser mittleren Farbe ergibt sich die (mittlere) Helligkeit der Kachel, wieder als Länge des Vektors. Die Helligkeit kann vorab für alle Kacheln gespeichert werden.
- Nun sucht man für jeden Pixel die Kachel mit der geringsten Helligkeitsabweichung und positioniert sie an der entsprechenden Stelle im Ergebnisbild.
- Dieses „best-fit“ Vorgehen führt bei Flächen gleicher Helligkeit zu unschönen Mustern, wenn Kacheln mehrfach verwendet werden. Auch werden bei kleinen Pools zum Ende immer weniger gut passende Kacheln gefunden, sodass die Qualität des Mosaiks dann z.B. nach unten rechts

abnimmt.

Eine deutliche Verbesserung ergibt sich durch eine Zufallsauswahl: man sucht erst die N besten Kacheln (z.B. N=20) und nimmt zufällig eine dieser N Kacheln.



Helligkeit :

$$|\vec{V}| = \sqrt{V_r^2 + V_g^2 + V_b^2}$$

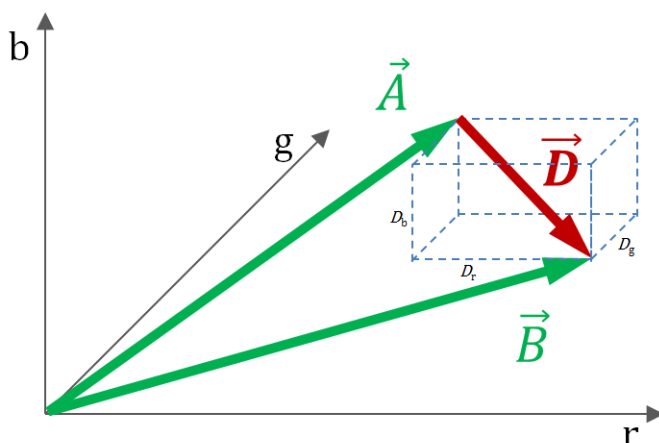
b) Kacheln mit geringstem Farbabstand suchen

Da beim Verfahren a) nur die Länge und nicht die Richtung der Farbvektoren berücksichtigt wird, passt z.B. eine hellrote Kachel ebenso wie eine hellgrüne oder helltürkise.

Um die Farbwerte (Richtungen der Vektoren) auch zu berücksichtigen, schlage ich vor, den „Farbabstand“ zwischen der Farbe des Basismotivpixels und der mittleren Kachelfarbe zu minimieren.

- Der Farbabstand ist die Länge des Vektors zwischen zwei Farbvektoren.
- Wie im Verfahren a) wird für alle Kacheln der mittlere Farbwert bestimmt und gespeichert.
- Für alle Basismotivpixel muss jetzt der Farbabstand zu allen Kacheln berechnet werden, um den kleinsten Abstand zu finden. Dies ist deutlich rechenintensiver, als beim ersten Verfahren.
- Wie bei a) führt auch hier ein „best-fit“ nicht unbedingt zu guten Ergebnissen. Per Zufall kann man auch hier aus den N Kacheln mit geringstem Farbabstand wählen.

Farbe eines Basismotivpixels: \vec{A}
mittlere Farbe einer Kachel: \vec{B}



$$\vec{A} + \vec{D} = \vec{B}$$

$$\vec{D} = \vec{B} - \vec{A}$$

$$D_r = B_r - A_r$$

$$D_g = B_g - A_g$$

$$D_b = B_b - A_b$$

Farbabstand: $|\vec{D}| = \sqrt{D_r^2 + D_g^2 + D_b^2}$