



HAL
open science

Extension des pseudo-gris pour la visualisation d'images scalaires sur des périphériques d'affichage couleur à faible profondeur de bits

David Tschumperlé, Patrick David, Olivier Lezoray

► To cite this version:

David Tschumperlé, Patrick David, Olivier Lezoray. Extension des pseudo-gris pour la visualisation d'images scalaires sur des périphériques d'affichage couleur à faible profondeur de bits. Colloque sur le Traitement du Signal et des Images (GRETSI'13), Sep 2013, Brest, France. pp.ID112. hal-00927554

HAL Id: hal-00927554

<https://hal.science/hal-00927554>

Submitted on 13 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Extension des pseudo-gris pour la visualisation d'images scalaires sur des périphériques d'affichage couleur à faible profondeur de bits.

David TSCHUMPERLÉ¹, Patrick DAVID² et Olivier LEZORAY¹

¹Laboratoire GREYC - Equipe IMAGE (UMR CNRS 6072)
6 Bd Maréchal Juin, 14050 Caen Cedex, France

²Affiliation of Pat David. Independant artist.

{David.Tschumperle,Olivier.Lezoray}@ensicaen.fr, patdavid@gmail.com

Abstract – We are interested in the visualization of *scalar* images - whose values are slightly quantized (or not at all) - on color displays which are highly quantized (typically 8-bits per channel). Nowadays, most of the available computer displays are like that, while image data become more and more accurate thanks to the increasing quality of image sensors (digital cameras for instance). We describe a simple solution in order to improve the visual quality of displayed gray-valued images. Our technique is inspired by the so-called *pseudo-gray* algorithm, initially proposed in [4]. We extend this method to limit the quantization effect of the displayed lightness as much as possible, thanks to the design of a more denser pseudo-gray colormap obtained under perceptual constraints.

1 Introduction - contexte

De nos jours, la plupart des périphériques d'affichage couleur standards (c-à-d l'ensemble *carte graphique + écran d'ordinateur*) se limitent à afficher des images dont les composantes rouge, verte et bleue de chaque pixel sont quantifiés sur 8 bits (256 valeurs possibles par composante). Dans le même temps, on assiste à une banalisation des capteurs capables d'acquérir des données images avec une précision de bits supérieure (c'est le cas par exemple des appareils photo numériques d'entrée de gamme), et de plus en plus de formats d'images savent stocker les valeurs des pixels avec des précisions d'au moins 12 bits, voire 16 bits par composante (format PNG ou RAW des appareils photos par exemple). Lors de l'affichage d'une image couleur à l'écran, cette quantification des valeurs des pixels est le plus souvent invisible à l'œil nu, puisqu'un affichage couleurs 8 bits/composante dispose d'une palette de plus de 16 millions de couleurs différentes. Mais si l'on se restreint spécifiquement à la visualisation d'*images scalaires* (c-à-d en niveaux de gris), et qu'on les représente en utilisant *des couleurs grises pures* (c-à-d de type $R = G = B = V$ où V est la valeur scalaire à représenter), on réduit la palette disponible pour l'affichage à 256 nuances différentes seulement. Pour un artiste professionnel spécialisé dans la photographie digitale noir et blanc et possédant un matériel d'impression perfectionné (capable d'imprimer avec une précision supérieure à 8 bits par composante), cette visualisation quantifiée est problématique, puisqu'elle peut introduire de légers artefacts visuels, peut-être invisibles sur l'impression finale, mais qui empêchent d'apprécier la justesse des post-traitements numériques appliqués sur l'image à haute précision. Pire encore, pour un photographe amateur désireux de produire une photo noir et blanc à partir d'une

photo couleur shootée en JPEG (8 bits par composante), le processus de quantification de la luminance se retrouve à la fois dans le stockage de la couleur dans le fichier image résultat (limitation du format JPEG) ainsi que dans la visualisation de la luminance à l'écran, alors que la luminance stockée pourrait être théoriquement moins quantifiée que ne le sont les couleurs de l'image d'entrée. Dans cet article, nous analysons une technique simple pour améliorer la qualité visuelle de la représentation des images scalaires à valeurs peu ou pas quantifiées (entiers 16 bits ou valeurs flottantes) sur des périphériques d'affichage couleurs, qui sont eux, davantage quantifiés (8 bits par composante par ex.). Nous nous basons sur une technique initiale de "vol de bit" [4, 5], reprise plus tard dans [1] sous l'appellation *pseudo-gris*. Nous étendons cette méthode pour générer des images couleurs qui ont (perceptuellement) l'apparence d'images en niveaux de gris, et qui permettent de diminuer drastiquement l'effet de quantification de la luminance perçue lors de l'affichage. Nous illustrons en particulier l'intérêt de cette visualisation pour une application de conversion d'images digitales couleurs en noir et blanc à partir de photos déjà quantifiées 8 bits (en format JPEG), pour l'affichage sur un écran à faible profondeur de bits.

2 Utilisation des pseudo-gris : technique initiale et extension proposée

Technique initiale : La méthode des pseudo-gris (abréviation : *PG*) permet de générer une représentation couleur d'une image scalaire haute précision (valeur entière de 16 bits ou plus) sur un périphérique d'affichage couleur, et est détaillée dans [1]. L'idée principale est de calculer l'image couleur visualisée

à partir de l'image d'origine haute précision, en utilisant une palette de couleurs qui, en apparence est en niveaux de gris, mais qui en réalité contient des pixels *légèrement colorés*. La couleur est ajoutée de manière suffisamment subtile pour que la coloration soit invisible à l'oeil nu, tout en permettant de générer de nouvelles nuances de "gris" intermédiaires (*pseudo-gris*) entre chaque couleurs grises pures (i.e., $R = G = B = V$). Ces nouvelles nuances dans la palette PG permettent donc d'étoffer l'ensemble des luminances affichables à l'écran. Elles sont calculées comme l'ensemble des couleurs possédant un incrément de +1 sur une ou plusieurs composantes (R, G, B) d'un gris pur (V, V, V), comme illustré sur la Figure.1.

(V, V, V)	(gris pur)	
$(V, V, V + 1)$	(pseudo-gris)	
$(V + 1, V, V)$	(pseudo-gris)	
$(V + 1, V, V + 1)$	(pseudo-gris)	
$(V, V + 1, V)$	(pseudo-gris)	
$(V, V + 1, V + 1)$	(pseudo-gris)	
$(V + 1, V + 1, V)$	(pseudo-gris)	
$(V + 1, V + 1, V + 1)$	(gris pur)	

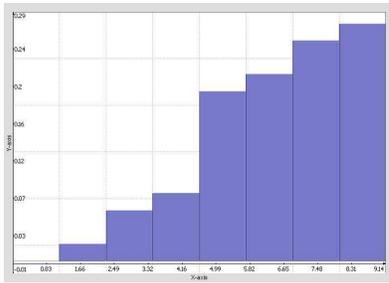


FIGURE 1 – Illustration de la technique originale des pseudo-gris. De gauche à droite : Séquence des couleurs PG insérées entre deux gris purs (en gras), visualisation des couleurs pseudo-grises et de leur luminance perçue (couleurs *normalisées*). En bas : Tracé de la luminance perçue.

Ces couleurs PG (représentées normalisées dans ce papier pour bien apprécier les différences de teintes), sont ordonnées par luminance perçue croissante (L de l'espace couleur Lab^*), dont on peut remarquer que la croissance n'est pas constante. On dispose au final d'une palette de $(255 \times 7 + 1) = 1786$ nuances de gris (plutôt que 256 gris purs) que l'on peut utiliser pour générer une représentation de l'image scalaire d'entrée à des fins de visualisation sur un écran couleur. On diminue ainsi l'effet de quantification de la luminance des niveaux de gris affichés, puisque l'on introduit des couleurs pseudo-grises intermédiaires ayant des luminances intermédiaires entre chaque gris pur. Pour le calcul de la représentation de l'image scalaire proprement dite, la technique initiale [1] propose de quantifier uniformément l'image d'entrée en 1786 niveaux, et à y mapper la palette de couleurs des 1786 pseudo-gris pour en obtenir une représentation directement visualisable à l'écran. C'est donc une méthode très facile à mettre en place, et peu coûteuse en temps de calcul.

Problèmes rencontrés et extension proposée : Quelques aspects de la technique originale des PG sont discutables :

1. Le mapping initial des couleurs PG se base sur une quantification *uniforme* de l'image scalaire à représenter. Mais la palette PG ne contient pas des couleurs dont la luminance perçue varie linéairement. Par conséquent, la luminance de la couleur d'une valeur mappée peut ne pas être la plus proche de celle de la valeur scalaire proprement dite (celle qui serait calculée sur le gris pur correspondant, en supposant une précision infinie du périphérique d'affichage). Nous proposons de minimiser cette erreur en mappant les couleurs de manière *non-linéaire*, de telle sorte que la différence de luminance perçue entre l'image d'origine et sa représentation PG soit minimale. Pour retrouver rapidement la couleur pseudo-grise optimale pour chaque valeur scalaire originale à mapper, nous pré-calculons en premier lieu une table d'association *luminance* \leftrightarrow *couleur pseudo-grise* qui contient suffisamment d'entrée pour contenir l'ensemble des valeurs possibles de l'image scalaire d'entrée (par exemple avec 2^{16} valeurs pour une image d'entrée quantifiée avec une précision de 16bits). Cette table donne donc un mapping direct de chaque valeur scalaire sous sa forme pseudo-grise quantifiée, en une complexité $O(1)$.

2. Nous proposons également d'autoriser plus d'incrément +1, +2, ..., + N sur les composantes couleurs d'un niveau de gris pur, afin d'obtenir encore plus de nuances de PG différentes. Plutôt que de déterminer des couleurs pseudo-grises intermédiaires entre chaque paire de gris purs (V, V, V) et $(V + 1, V + 1, V + 1)$, nous considérons le problème d'un point de vue plus global, en générant l'ensemble des triplets couleurs $(V + dR, V + dG, V + dB)$, où $dR, dG, dB \in \mathbb{N}$ peuvent prendre les valeurs de 0 à + N . Cet ensemble est ensuite ordonné par luminance croissante pour obtenir une palette PG_N de couleurs pseudo-grises étendue. Notons que ces palettes étendues vérifient $PG_N \subset PG_{N+1}$ et que la palette PG_1 correspond effectivement à celle obtenue par la formulation PG d'origine [1] (à 1786 entrées). Augmenter N permet d'obtenir plus d'entrées dans les palettes PG obtenues et participe donc à l'amélioration de la précision visuelle de la représentation des images scalaires : Les différences des luminances des couleurs affichées par rapport aux luminances exactes deviennent de plus en plus faibles et très peu quantifiées, au fur et à mesure que N augmente (Figure.2).

3. Bien sûr, lorsque N devient grand, on obtient des teintes pseudo-grises de plus en plus colorées, et on risque de s'apercevoir de ces couleurs dans PG_N . Nous introduisons donc une contrainte perceptuelle basée sur la mesure Δ_E de différence entre couleurs [3] pour contraindre la palette des pseudo-gris générée à contenir des couleurs qui sont suffisamment proches (dans le sens perceptuel) d'un gris pur, tout en contraignant également deux couleurs adjacentes d'une palette PG_N à être perceptuellement indistinguables (Δ_E inférieur au seuil JND, *Just Noticeable Difference*). On assure ainsi que les couleurs de

PG_N restent pseudo-grises, tout en atténuant les discontinuités perceptuelles dans le dégradé de pseudo-gris généré. Cette contrainte est mise en place de la façon suivante :

- Calcul de la palette PG_N , pour un incrément N donné. Cette palette contient généralement un grand nombre de couleurs.
- Suppression dans PG_N de toutes les couleurs $C = (L, a, b)$ perceptuellement trop éloignées des niveaux de gris “idéaux” (considérés ayant une précision infinie) qui auraient la même luminance que C , c-à-d les entrées telles que $\sqrt{a^2 + b^2} > \text{JND}$.
- Suppression dans PG_N des couleurs indicées par i ne vérifiant pas $\Delta_E(PG_N(i), PG_N(i+1)) < \text{JND}$. Cette dernière étape est répétée jusqu’à convergence de la palette.

Les points 2 et 3 de l’extension des pseudo-gris que nous proposons sont antagonistes, puisque l’on souhaite à la fois augmenter le nombre de teintes PG disponibles, et contraindre perceptuellement les couleurs. Pour une valeur de JND donnée, il existe ainsi un N optimal tel que le nombre d’entrées dans la palette PG_N calculée soit maximale et que les contraintes perceptuelles soient respectées. Par exemple, pour un JND de 2.3, la génération des différentes palettes PG_N sous contraintes amènent au meilleur compromis obtenu pour $N = 5$, pour une palette $PG_5^{\text{JND}=2.3}$ générée qui contient finalement 8905 teintes de couleurs pseudogrises disponibles (contre 1786 pour la méthode des PG initiale). Si l’on resserre la contrainte en considérant par exemple un JND à 1, on baisse significativement le nombre d’entrées dans les palettes PG_N obtenues, et le meilleur compromis est alors atteint pour $N = 2$, avec une palette $PG_2^{\text{JND}=1}$ contenant 1421 couleurs pseudogrises. Dans ce cas plus strict, notons que la formulation initiale PG_1 proposée dans [1] ne vérifie pas les contraintes perceptuelles.

3 Applications

Nous nous intéressons ici à une des applications pratiques de notre méthode, à savoir la conversion d’une image couleur en une image en niveaux de gris, dans un but de visualisation sur un périphérique d’affichage couleur à K bits de quantification par composante (ici un écran ou une imprimante). Notons que l’image couleur d’entrée peut elle-même être quantifiée (8bits par composante par exemple), mais nous considérons que l’image scalaire résultante que nous cherchons à visualiser ne l’est pas, ou très peu. Ceci est vrai pour notre application puisque notre image scalaire est calculée comme la luminance de l’image couleur d’entrée, et que ce calcul de mélange des composantes couleurs *déquantifie* donc de manière suffisante le signal d’entrée. La méthode que nous proposons dans ce papier permet d’atténuer l’effet de quantification de luminance perçue lors de l’affichage d’une image scalaire sur un périphérique d’affichage quantifié, en ajoutant des teintes pseudo-grises intermédiaires. Néanmoins, pour un affichage traditionnel de 8bits par composante, il est évident que cette quantification de la luminance est déjà relativement imperceptible pour un oeil non averti. Pour rendre notre approche tangible, nous proposons

donc de l’illustrer par deux cas $K = 8\text{bits}$ (pour une image synthétique) et $K = 4\text{bits}$ (pour une image réelle).

La Figure.2 illustre le cas synthétique suivant : une image couleur à convertir contient un dégradé de couleurs cyan-vert, quantifié en 8 bits par composante. Une méthode usuelle de conversion en niveaux de gris consiste à générer pour chaque pixel une couleur $R = G = B = Y$, où Y représente la luminance dans l’espace couleur YUV , ce que nous faisons ici. Les différents dégradés de la Figure.2 ont été représentés normalisés (étirés sur $[0, 255]$) pour mieux rendre compte des variations de luminance et de couleur des images grises ou pseudo-grises générées pour la visualisation. Il apparaît effectivement que la visualisation (B) en niveaux de gris purs (sur 8 bits) quantifie de manière importante la luminance perçue attendue lors de l’affichage. L’utilisation de palettes PG_1 (C) et PG_5 (D) permettent quant à elles de générer des dégradés pseudo-gris dont la luminance perçue est effectivement plus proche de la luminance “idéale” (A) de l’image scalaire à haute précision. Il est difficile d’apercevoir nettement à l’écran les effets d’escaliers dû à la visualisation quantifiée (B), mais on perçoit une plus grande régularité globale des dégradés de teintes dans les cas (C) et (D), et la sensation que l’image est plus “lisse” (alors même qu’aucun filtrage spatial n’a été appliqué). A noter qu’ici, le cas (C) a été généré par mapping non-linéaire de la palette PG_1 , et correspond donc à notre extension proposée et non à la méthode PG initiale telle que proposée dans [4, 5]. Par ailleurs, notre méthode PG_5 n’introduit aucun surcoût en temps machine par rapport à PG_1 .

La Figure.3 illustre la visualisation d’une image réelle sur un périphérique d’affichage à 4bits de profondeur. L’utilisation de notre algorithme PG permet d’utiliser une palette de couleur pseudo-grises enrichie de 45 nuances (au lieu des 16 gris purs théoriques). Notre extension des PG est suffisamment générique pour traiter ces différentes profondeurs de bits de manière transparente.

Références

- [1] R. Franzen, “PseudoGrey : Squeezing 1786 Grey Levels out of 256 Possibilities”, URL <http://r0k.us/graphics/pseudoGrey.html>, Jul. 1999.
- [2] H. Kikuchi, H. Sato, S. Hasebe, N. Mizutani, S. Muramatsu, S. Sasaki, J. Zhou, S. Sekine, Y. Abe, I. Tofukuji and M. Nakashizuka, “Extra encoding of fine grayscale data into 8-bit sRGB color space”, Journal of Optical Engineering, Vol.42, No.7, pp.1940-1948, 2003.
- [3] A.R. Robertson, “Historical development of CIE recommended color difference equations”, Proc. Journal of Color Research & Application, Vol.15, No.3, pp. 167–170, June 1990.
- [4] C.W. Tyler, H. Chan, L. Liu, B. McBride and L.L. Kontsevich, “Bit-stealing : How to get 1786 or more grey levels from an 8-bit color monitor”, Proc. SPIE 1666, Human Vision, Visual Processing, and Digital Display III, Vol.351, pp. 351–364, August. 1992.
- [5] C.W. Tyler, “Colour bit-stealing to enhance the luminance resolution of digital displays on a single pixel basis”, Journal of Spatial Vision, Vol.10, No.4, pp. 369–377, 1997.

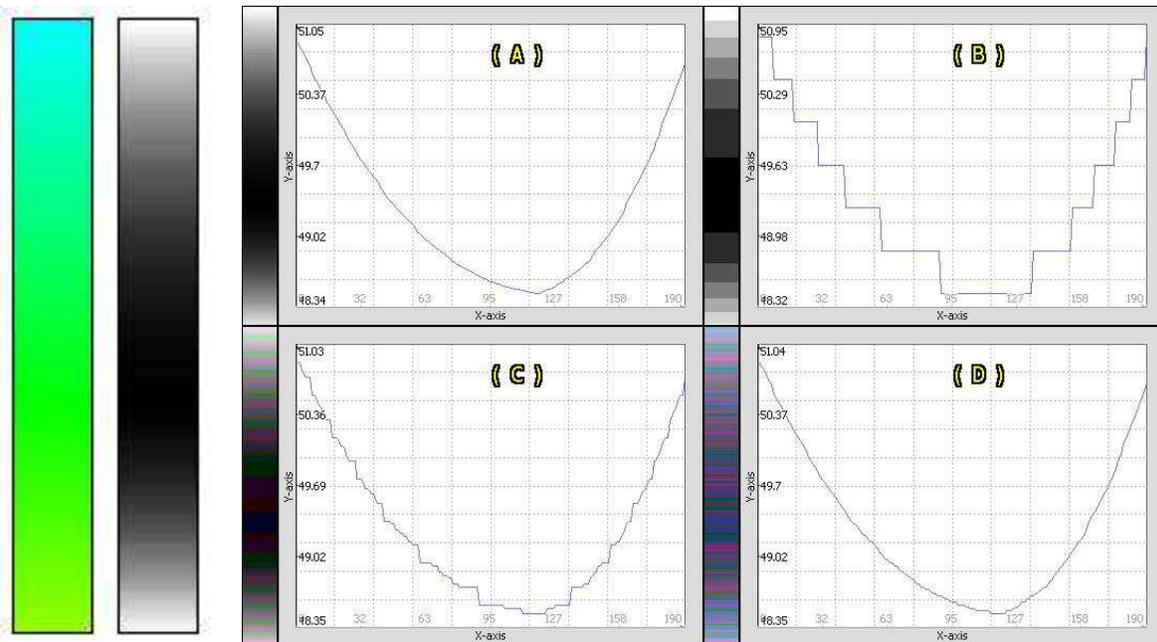


FIGURE 2 – Illustration de l’extension des pseudo-gris pour la conversion d’un dégradé de couleurs en niveaux de gris, et sa visualisation. De gauche à droite : Dégradé couleur à convertir (**normalisé**), luminance exacte correspondante (**normalisée**), (A) tracé de la luminance perçue idéale, (B) luminance perçue en utilisant des niveaux de gris purs pour la visualisation, (C) luminance perçue en utilisant une palette PG_1 mappée non-linéairement, (D) luminance perçue en utilisant une palette PG_5 sous contrainte perceptuelle, mappée non-linéairement.

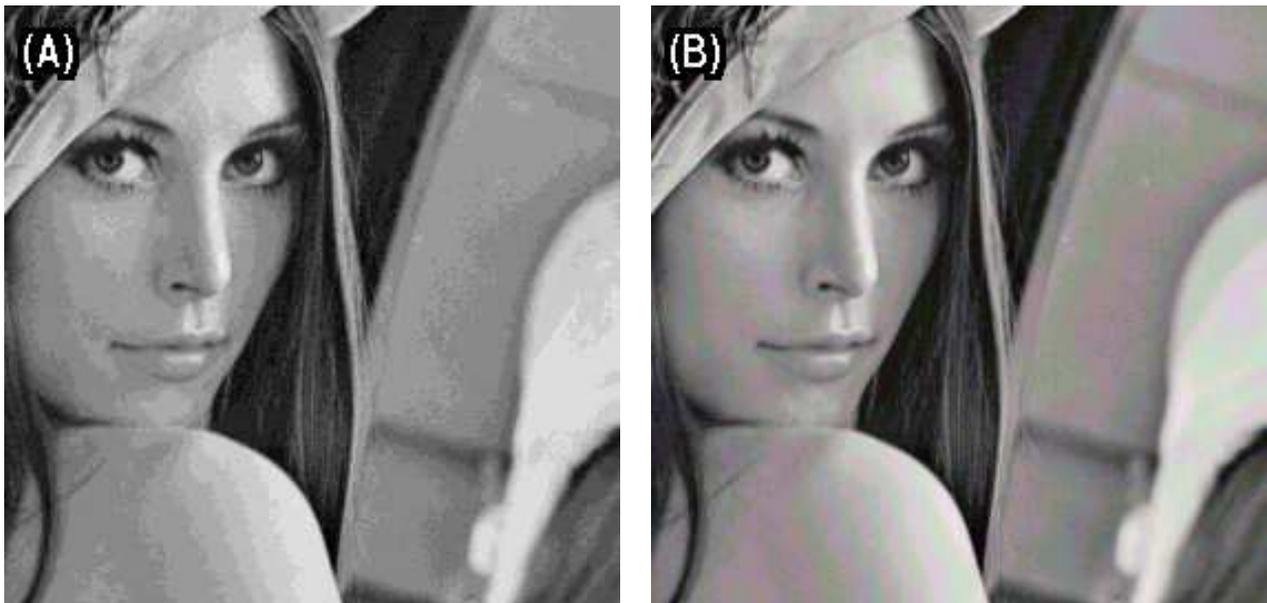


FIGURE 3 – Représentation d’une image scalaire sur un périphérique d’affichage couleur à 4bits/composantes. (A) image scalaire utilisant des niveaux de gris purs pour la visualisation (16 niveaux), (B) image scalaire obtenue en mappant une palette $PG_2^{JND=5}$, contenant 45 entrées. Le JND a été augmenté ici pour que les couleurs pseudogris à 4bits/composantes puissent vérifier les contraintes perceptuelles.