



HAL
open science

Stabilisation tonale de vidéos

Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier

► **To cite this version:**

Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier. Stabilisation tonale de vidéos. GRETSI 2015, Sep 2015, Lyon, France. hal-01256950

HAL Id: hal-01256950

<https://hal.science/hal-01256950v1>

Submitted on 25 Jan 2016

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.

Stabilisation tonale de vidéos

Oriel FRIGO^{1,2}, Neus SABATER¹, Julie DELON², Pierre HELLIER¹

¹Technicolor, Rennes, France

²Université Paris Descartes, MAP5, Paris, France

oriel.frigo@technicolor.com,

neus.sabater@technicolor.com, julie.delon@parisdescartes.fr, pierre.hellier@technicolor.com

Résumé – Dans ce travail, nous proposons une méthode rapide et paramétrique pour corriger l’instabilité tonale de vidéos. La méthode utilise un modèle de correction colorimétrique minimal mais efficace, et s’appuie sur une estimation paramétrique du mouvement dominant. La correction est pondérée temporellement, en fonction de l’amplitude du mouvement. Les expériences menées sur différentes séquences montrent que cette méthode dépasse les performances de l’état de l’art, à la fois en terme de qualité visuelle et de temps de calcul.

Abstract – In this work, we present a fast and parametric method to achieve tonal stabilization in videos containing color fluctuations. Our main contribution is to compensate tonal instabilities with a color transformation guided by dominant motion estimated between temporally distant frames. Furthermore, we propose a temporal weighting scheme, where the intensity of tonal stabilization is directly guided by the motion speed. Experiments show that the proposed method compares favorably with the state-of-the-art in terms of accuracy and computational complexity.

1 Introduction

Les caméras grand public, et en particulier les smartphones, règlent généralement la balance des blancs ainsi que l’exposition des vidéos de manière automatique, ces corrections n’étant pas toujours désactivables. Malheureusement, ces réglages sont rarement stables dans le temps, et provoquent des fluctuations de tonalité et d’exposition. Ces fluctuations sont non seulement gênantes visuellement, mais posent également problème à toutes les approches de vision par ordinateur faisant l’hypothèse d’une stabilité temporelle. Nous proposons dans cet article une méthode automatique pour corriger ces instabilités *a posteriori*.

D’après les auteurs de [12], l’image u enregistrée par une caméra est liée à l’image brute (RAW) de capteur obtenue après l’étape de désentrelacement par l’équation :

$$u = f \circ h(\mathbf{T}_s \mathbf{T}_w \mathbf{e}), \quad (1)$$

où \mathbf{T}_s est une matrice 3×3 de changement d’espace couleur, \mathbf{T}_w est une matrice diagonale 3×3 de balance des blancs, $h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ est un opérateur non-linéaire dit de *gamut mapping* et $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ est une transformation tonale non-linéaire. La balance des blancs est contrôlée par \mathbf{T}_w , tandis que l’exposition peut être assimilée à un facteur multiplicatif sur le vecteur \mathbf{e} . Les fluctuations liées à ces deux facteurs sont donc des transformations globales sur chaque image de la vidéo.

Si la transformation non-linéaire $f \circ h$ était connue, il serait possible après inversion de corriger la séquence $\{(f \circ h)^{-1}(u_t)\}_t$ par des transformations linéaires 3D. Malheureusement, pour estimer la fonction de transfert f de la caméra et le gamut mapping h , il est nécessaire d’utiliser des images recalées acquises

avec différentes expositions [9], et des paires RAW-sRGB [12], ce qui n’est pas concevable pour une utilisation à grande échelle. De même, reposer sur une méthode d’estimation d’illuminants [8] pour corriger ces instabilités tonales serait trop instable en pratique. Nous proposons dans cet article une méthode générique de stabilisation ne faisant aucune hypothèse, ni sur l’illuminant de la scène, ni sur le modèle de caméra.

Alors que peu de travaux sont consacrés à la correction d’instabilité tonale, de nombreux articles abordent le problème des fluctuations de niveaux de gris (*flicker*) dans les films, proposant des approches de correction globales [2, 3] ou locales [14, 4]. Malheureusement, l’extension de ces méthodes à des fluctuations colorimétriques n’est pas triviale. De manière orthogonale, si le problème du transfert de couleur entre images intéresse de nombreux auteurs [15, 6, 7], les solutions proposées sont trop coûteuses en pratique pour les envisager dans le cadre de la correction tonale.

A notre connaissance, seuls deux articles proposent des solutions spécifiques à l’instabilité tonale [5, 16]. L’approche de Farbman *et al* [5] tente d’aligner la tonalité de chaque image sur celle d’une ou plusieurs images choisies de la vidéo (par exemple, la première image). Les correspondances entre images sont obtenues sans compensation de mouvement. Cette méthode, dont les temps de calcul sont assez importants, fonctionne bien sur des séquences statiques, mais les résultats sont moins probants sur des séquences bruitées, en mouvement, ou de longue durée. L’approche plus récente de Wang *et al*. [16] s’appuie sur une estimation paramétrique du mouvement entre images. Ces transformations sont enchaînées afin de définir un état colorimétrique, qui est lissé dans le temps. Si les résultats sont visuellement convaincants, cette approche reste complexe, dif-

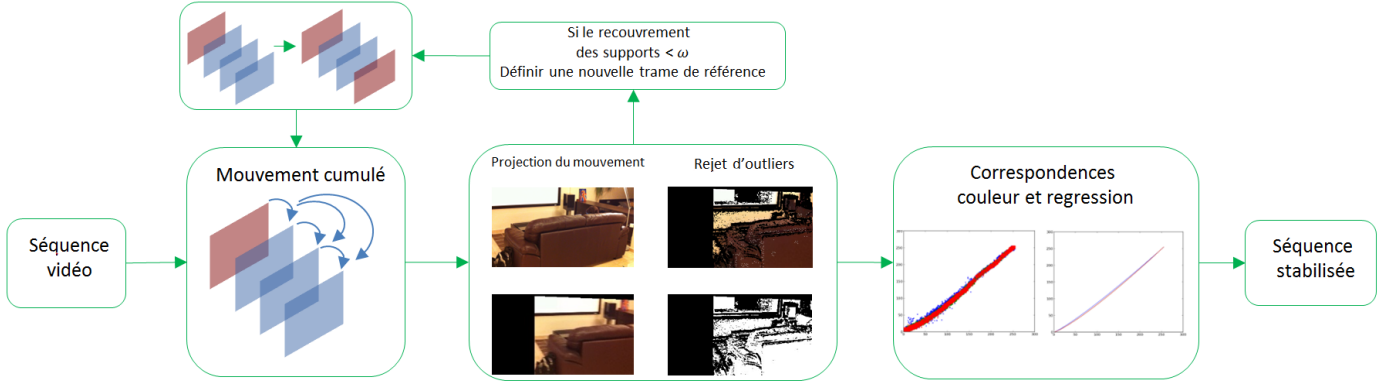


FIGURE 1 – Schéma général de la méthode de stabilisation tonale.

ficile à réimplémenter, et nécessite le réglage fin de paramètres.

Notre méthode, présentée sous forme schématique dans la Figure 1 et détaillée dans la section 2, s’appuie sur un modèle simplifié d’instabilité couleur entre images. Grâce à une estimation paramétrique du mouvement dominant, elle est robuste au mouvement et aux occultations, et adaptée à une implémentation temps-réel embarquée. La section 3 illustre l’efficacité de la méthode sur plusieurs séquences. Les résultats obtenus s’avèrent supérieurs à ceux de [5] et qualitativement équivalent à ceux de [16], mais avec un coût de calcul considérablement réduit. L’ensemble des résultats est disponible en ligne ¹.

2 Méthode de stabilisation tonale

Cette section décrit notre méthode de stabilisation tonale. Nous présentons d’abord le modèle de correction colorimétrique pour une séquence sans mouvement, puis sa généralisation aux séquences en mouvement.

2.1 Modèle de correction colorimétrique

Notons $\{u_t\}_{t=1,\dots,D}$ une séquence recalée d’images couleur $u_t : \Omega \rightarrow \mathbb{R}^3$, où $\Omega \subseteq \mathbb{R}^2$ est le domaine spatial de chaque image. Les composantes couleur de u_t sont notées (u_t^R, u_t^G, u_t^B) . Notons u_k une image de référence dans la séquence. Pour stabiliser l’apparence couleur de la séquence, pour toute image $u_t, t > k$ postérieure à u_k , nous cherchons une transformation colorimétrique $T_t : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ telle que $T_t(u_t) \simeq u_k$. Nous utilisons ici un modèle volontairement simple, séparable selon les trois composantes :

$$T_t = (T_t^R, T_t^G, T_t^B), \quad \text{où } T_t^c(s) := \alpha_c s^{\gamma_c}, \quad c \in \{R, G, B\}. \quad (2)$$

Ce modèle, très simple, suffit en pratique pour corriger les instabilités tonales au sein d’une vidéo. Pour s’en assurer, nous l’avons testé sur un ensemble d’images de la même scène, en faisant varier la balance des blancs et l’exposition. Dans tous les cas, ce modèle s’est révélé suffisamment précis pour corriger les fluctuations observées.

Afin d’estimer les paramètres α_c, γ_c pour l’image u_t , on résout le problème aux moindres carrés suivant pour chaque canal couleur :

$$\arg \min_{\alpha_c, \gamma_c} \sum_{\mathbf{x} \in \Omega} (\log u_k^c(\mathbf{x}) - \gamma_c \log u_t^c(\mathbf{x}) + \log \alpha_c)^2. \quad (3)$$

Résoudre ce problème n’est évidemment pas équivalent à résoudre directement le problème aux moindres carrés sur la différence $u_k - \alpha u_t^{\gamma_c}$, mais présente l’avantage de posséder une solution analytique calculable très rapidement :

$$\gamma_c = \frac{\text{Cov}(\log u_k^c, \log u_t^c)}{\text{Var}(\log u_t^c)}, \quad \alpha_c = \exp(\overline{\log u_k^c} - \gamma_c \overline{\log u_t^c}), \quad (4)$$

où $\bar{z} = \frac{1}{|\Omega|} \sum_{\mathbf{x} \in \Omega} z(\mathbf{x})$ est la moyenne de $z(\mathbf{x})$, $\mathbf{x} \in \Omega$.

2.2 Compensation de mouvement

Nous utilisons la méthode robuste de Odobez et Boutheymy [13] pour calculer un mouvement affine $A_{l,l-1}$ (à 6 paramètres) pour chaque paire d’images consécutives u_l et u_{l-1} . Cette correction affine du mouvement, si elle ne peut compenser que le mouvement dominant de caméra, présente l’avantage d’être très simple à calculer, et nous verrons qu’elle est suffisante pour la correction d’instabilités tonales dans les vidéos.

Etant données une image de référence u_k et une image postérieure u_t ($t > k$) dans la séquence, u_t est recalée vers u_k en utilisant la transformation cumulée :

$$A_{t,k} = A_{t,t-1} \circ A_{t-1,t-2} \circ \dots \circ A_{k+1,k}. \quad (5)$$

Nous définissons ensuite l’ensemble $\Omega_{t,k}$ des correspondances spatiales entre u_k et u_t de la manière suivante :

$$\Omega_{t,k} = \left\{ (\mathbf{x}, A_{t,k}(\mathbf{x})) \in \Omega \times \Omega \mid \frac{1}{3} \sum_c \left((u_k^c(\mathbf{x}) - \bar{u}_k^c) - (u_t^c(A_{t,k}(\mathbf{x}))) - \bar{u}_t^c \right)^2 < \sigma^2 \right\}, \quad (6)$$

où σ^2 est la variance empirique du bruit, estimée par exemple en utilisant [1]. Cette définition des correspondances spatiales

1. http://oriel.github.io/tonal_stabilization

est robuste aux *outliers* de mouvement, ainsi qu'aux point occultés, c'est à dire visibles uniquement dans une seule des deux images.

2.3 Algorithme

En pratique, la première image de la séquence est choisie comme première image de référence. Chaque image postérieure $u_t, t > 1$ est ensuite corrigée tonalement en lui appliquant la transformation T_t^c dont les paramètres sont estimés en résolvant le problème (3) entre u_1 et l'image recalée $A_{t,1}u_t$. Les images successives de la séquence sont ainsi corrigées tant que le cardinal de l'ensemble des correspondances entre u_t et u_1 est suffisamment grand, *i.e.* tant que $|\Omega_{t,1}| > \omega \times |\Omega|$, où ω est un paramètre à régler. Si le nombre de correspondances spatiales devient trop faible pour estimer une bonne transformation couleur, une nouvelle image de référence est définie. Ce processus est répété jusqu'à la fin de la séquence.

Pour garantir une forme de fidélité entre la séquence finale et la séquence originale $\{u_t\}_{t=1,\dots,D}$, nous introduisons une pondération temporelle similaire à un terme de viscosité :

$$T_t' = \lambda T_t + (1 - \lambda)Id, \quad (7)$$

où $\lambda = \lambda_0 \exp(-\frac{\|V_{t,k}\|}{p})$ régle le niveau de transformation de u_t , en fonction de l'amplitude de mouvement $\|V_{t,k}\|$, où $V_{t,k}$ est le vecteur de mouvement dominant, p est le déplacement spatial maximal (dimension de l'image), et λ_0 est le paramètre de pondération initial (en pratique, $\lambda_0 = 0.9$). Nous avons constaté que cette correction stabilise l'apparence tonale tout en évitant une sur-exposition lorsque l'exposition de la caméra change au cours de la séquence. Le détail de l'algorithme proposée est présenté ci-dessous.

Algorithm 1 Stabilisation tonale de vidéos

Données d'entrée : Séquence vidéo $\{u_t\}_{t=1,\dots,D}$

Données de sortie : Séquence vidéo stabilisée

```

1:  $\{T_t'(u_t)\}_{t=1,\dots,D}$ 
2:  $k \leftarrow 1, t \leftarrow k + 1$ 
3:  $T_1'(u_1) = u_1$ 
4: while  $t \leq D$  do
5:   Calculer  $\Omega_{t,k}$ 
6:   if  $\#\Omega_{t,k} \geq \omega \times |\Omega|$  then
7:     for  $c \in \{r, g, b\}$  do
8:        $\alpha_c, \gamma_c \leftarrow \arg \min_{\alpha, \gamma} \sum_{(\mathbf{x}, \mathbf{y}) \in \Omega_{t,k}} (u_k^c(\mathbf{x}) - \alpha u_t^c(\mathbf{y})^\gamma)^2$ 
9:        $T_t'(u_t^c) \leftarrow \lambda \alpha_c (u_t^c)^{\gamma_c} + (1 - \lambda) u_t^c$ 
10:    end for
11:     $t \leftarrow t + 1$ 
12:  else # Si le nombre de correspondances est trop faible
13:     $k \leftarrow t - 1$ 
14:     $u_k \leftarrow T_{t-1}'(u_{t-1})$ 
15:  end if
16: end while

```

Pour réduire la complexité de l'algorithme, les images originales sont redimensionnées à 120 pixels de largeur avant l'es-

timation de T' . La transformation T' est encodée comme une *Look-Up-Table (LUT)* avant d'être appliquée aux images à pleine résolution. Cet algorithme produit une bonne stabilisation tonale avec une complexité calculatoire maîtrisée.

3 Résultats expérimentaux

La méthode proposée a été testée sur un ensemble de séquences provenant de [5, 16] ou ayant été acquises avec des smartphones de différentes marques. L'ensemble des résultats peut être consulté sur la page web du projet². Des exemples de résultats obtenus sur deux séquences sont présentés sur les Figures 2 et 3. Pour toutes les séquences, nous fixons les paramètres à $\omega = 0.25$ et $\lambda_0 = 0.9$.

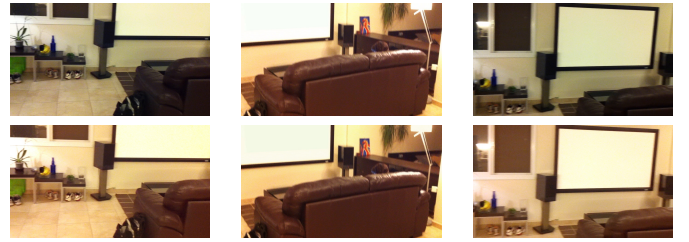


FIGURE 2 – Résultats sur la séquence *sofa* (fournie par les auteurs de [5]). Première ligne : images extraites de la séquence initiale. Seconde ligne : images de la vidéo corrigée par notre méthode de stabilisation tonale.

Afin de comparer quantitativement nos résultats aux méthodes [5, 16], nous avons re-implémenté la méthode [5], et les auteurs de [16] nous ont fourni leurs résultats sur les séquences considérées. Pour mesurer la stabilité des résultats obtenus, un patch homogène est défini et suivi sur la séquence, et nous calculons la variation temporelle de la tonalité du patch après stabilisation, mesurée par une distance euclidienne dans l'espace couleur CIELAB. Les résultats sur la séquence *bâtiment* sont visibles sur la Figure 4. Sur les courbes de gauche, observons que la variation tonale du patch est réduite après stabilisation par notre méthode et celle de Wang *et al.* [16], mais que la méthode de Farbman *et al.* [5] donne naissance à quelques fluctuations résiduelles. Les courbes de droite illustrent la fidélité aux couleurs initiales, mesurée par la distance couleur entre chaque image corrigée et l'image originale, afin de vérifier que les méthodes de stabilisation ne produisent pas de distorsions. Sur ces différentes courbes, nous pouvons voir que notre méthode produit des résultats compétitifs, tant en terme de stabilisation que de fidélité aux couleurs originales.

Notre approche est par ailleurs beaucoup plus rapide que les méthodes précédentes. Un prototype implémenté en langage Python peut traiter 20 images par seconde³ pour une vidéo HD 1920 · 1080. , alors que l'implémentation C++ de [16] atteint une cadence de 1 image par seconde, et l'implémentation Py-

2. http://oriel.github.io/tonal_stabilization

3. Temps de calcul sur une architecture Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz, 8GB RAM



FIGURE 3 – Comparaison visuelle des résultats sur la séquence *bâtiment*. Notre méthode stabilise l'apparence tonale des objets sans distorsions colorimétrique, tandis que les méthodes [5, 16] génèrent quelques artefacts de saturation sur la façade du bâtiment.

thon de [5] est à 0.6 image par seconde. Nous pensons qu'une implémentation optimisée pourrait ainsi être temps-réel sur une application embarquée.

4 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode pour stabiliser l'apparence tonale de vidéos, reposant sur une estimation du mouvement dominant de la caméra et une correction colorimétrique paramétrique. Nous avons montré que cette correction colorimétrique très simple permet une stabilisation efficace, sans connaissance *a priori* sur le modèle de caméra. Les bénéfices de cette approche sont multiples : d'une part, cette méthode est robuste vis-à-vis des séquences contenant du mouvement, d'autre part la complexité calculatoire est très réduite, ce qui ouvre la voie aux applications temps-réel embarquées.

Remerciements : Nous remercions les auteurs de la référence [16], qui nous ont très gentiment fourni les résultats de leur méthode sur plusieurs séquences.

Références

- [1] M. Colom and A. Buades. Analysis and Extension of the Percentile Method, Estimating a Noise Curve from a Single Image. *IPOLE*, 3 :332–359, 2013.
- [2] E. Decencière. *Restauration automatique de films anciens*. Thèse, Décembre 1997.
- [3] J. Delon. Movie and video scale-time equalization application to flicker reduction. *IEEE TIP*, 15(1) :241–248, Jan 2006.

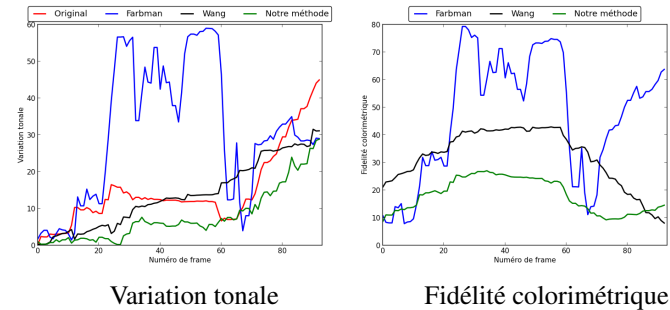


FIGURE 4 – Résultats quantitatifs sur la séquence *bâtiment*, présentée en Fig. 3. A gauche, la stabilité tonale est calculée sur un patch homogène suivi le long de la séquence, puis mesurée par la distance couleur sur ce patch. A droite, la fidélité couleur est calculée par la distance colorimétrique entre chaque image et l'image corrigée. En particulier, l'apparence trop blanche de la façade pour la méthode [16] se traduit par une fidélité dégradée. De manière générale, notre méthode est très compétitive par rapport à l'état de l'art, tant en terme de résultats quantitatifs de stabilisation, de fidélité aux couleurs originales, qu'en terme de temps de calcul.

- [4] J. Delon and A. Desolneux. Stabilization of flicker-like effects in image sequences through local contrast correction. *SIAM J. on Imaging Sciences*, 3(4) :703–734, 2010.
- [5] Z. Farbman and D. Lischinski. Tonal stabilization of video. *SIGGRAPH*, 30(4) :89 :1 – 89 :9, 2011.
- [6] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J. F. Aujol. Regularized discrete optimal transport. *Scale Space and Variational Methods in Computer Vision*, 428–439, 2013.
- [7] O. Frigo, N. Sabater, V. Demoulin, and P. Hellier. Optimal transportation for example-guided color transfer. In *ACCV*, 2014.
- [8] A. Gijssenij, T. Gevers, and J. Van De Weijer. Computational color constancy : Survey and experiments. *IEEE TIP*, 20(9) :2475 – 2489, 2011.
- [9] M. Grossberg and S. Nayar. What Can Be Known about the Radiometric Response from Images ? In *ECCV*, pages 189–205. 2002.
- [10] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *SIGGRAPH*, 30(4) :70 :1–70 :10, July 2011.
- [11] S. J. Kim, H. T. Lin, Z. Lu, S. Susstrunk, S. Lin, and M. S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE PAMI*, 34 :2289–2302, 2012.
- [12] H. Lin, S. J. Kim, S. Susstrunk, and M. S. Brown. Revisiting radiometric calibration for color computer vision. In *ICCV*, pages 129–136, 2011.
- [13] JM. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. of visual communication and image representation*, 6(4) :348–365, 1995.
- [14] F. Pitié, R. Dahyot, F. Kelly, and A. Kokaram. A new robust technique for stabilizing brightness fluctuations in image sequences. In *Statistical Methods in Video Processing*, pages 153–164. Springer, 2004.
- [15] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5) :34–41, sep/oct 2001.
- [16] Y. Wang, D. Tao, X. Li, M. Song, J. Bu, and P. Tan. Video tonal stabilization via color states smoothing. *IEEE TIP*, 23(11) :4838–4849, 2014.