

YANIS CADI · STUDIO

# SITES 3D CINÉMATIQUES

La méthode complète pour créer un site où la caméra traverse le lieu au fil du scroll, comme dans un film. Des images au site en ligne.

01 LA MÉTHODE

02 LES OUTILS PRÊTS À L'EMPLOI

03 LA VENTE

# CE GUIDE, C'EST MON ATELIER OUVERT

Tu as vu passer mes vidéos : un site qui se visite comme un film, la caméra qui avance dans un couloir, entre dans un écran, plonge dans un café jusqu'à la tasse. Beaucoup m'ont demandé comment je fais. Voici tout.

Ce guide n'est pas une théorie. C'est la méthode que j'utilise pour mes propres sites, avec les outils que j'ai écrits et testés sur de vrais projets. Chaque commande de ce guide a été lancée et vérifiée avant d'être imprimée ici.

Tu vas apprendre à fabriquer l'effet de A à Z : écrire le voyage de la caméra, générer des plans qui s'emboîtent parfaitement, les monter en un film continu, et le faire défiler au rythme du scroll sur une page web qui charge vite, sur ordinateur comme sur téléphone.

## CE QUE TU SAURAS FAIRE

- ▶ Concevoir un storyboard « poupée russe »
- ▶ Générer des plans cohérents avec l'IA
- ▶ Passer tes images en haute définition
- ▶ Monter le film avec mon script
- ▶ Construire la page qui défile au scroll
- ▶ Mettre en ligne et vendre le résultat

## CE QU'IL TE FAUT

- ▶ Un ordinateur (Mac, Windows ou Linux)
- ▶ Un générateur d'images qui accepte une image de référence (ChatGPT par exemple)
- ▶ Python 3 et ffmpeg, gratuits
- ▶ Un compte Netlify, gratuit
- ▶ Un week-end pour ton premier site

## AUCUN NIVEAU DE CODE REQUIS

Tu n'as pas besoin de savoir programmer. Les trois outils sont fournis prêts à l'emploi : tu copies, tu lances une commande, tu obtiens ton film et ta page. Je t'explique quand même ce que fait chaque ligne importante, pour que tu puisses adapter.

# LE VOYAGE, PLAN PAR PLAN

|           |                                                     |       |
|-----------|-----------------------------------------------------|-------|
| <b>01</b> | <b>CE QU'EST UN SITE 3D CINÉMATIQUE</b>             | p. 04 |
|           | L'effet, pourquoi il fonctionne, ce qu'il n'est pas |       |
| <b>02</b> | <b>LE PRINCIPE DE LA POUPÉE RUSSE</b>               | p. 07 |
|           | La règle d'or qui rend le zoom continu possible     |       |
| <b>03</b> | <b>ÉCRIRE LE STORYBOARD</b>                         | p. 10 |
|           | Du plan large au détail qui donne envie             |       |
| <b>04</b> | <b>GÉNÉRER LES PLANS AVEC L'IA</b>                  | p. 13 |
|           | Prompts, cohérence, plans intermédiaires            |       |
| <b>05</b> | <b>TRAVAILLER AVEC DE VRAIES PHOTOS</b>             | p. 17 |
|           | Clients, personnes, ressemblance, honnêteté         |       |
| <b>06</b> | <b>LA HAUTE DÉFINITION</b>                          | p. 20 |
|           | Agrandir sans flou, sans effet plastique            |       |
| <b>07</b> | <b>MONTER LE FILM</b>                               | p. 23 |
|           | Mon script film_zoom.py, pas à pas                  |       |
| <b>08</b> | <b>ENCODER POUR LE WEB</b>                          | p. 27 |
|           | Fluide au scroll, léger au chargement               |       |
| <b>09</b> | <b>LA PAGE QUI FAIT DÉFILER LE FILM</b>             | p. 30 |
|           | Mon modèle page_scroll.html expliqué                |       |
| <b>10</b> | <b>METTRE EN LIGNE</b>                              | p. 33 |
|           | Netlify en glisser-déposer, nom de domaine          |       |
| <b>11</b> | <b>LES PIÈGES QUE J'AI PAYÉS</b>                    | p. 34 |
|           | Huit erreurs et comment les éviter                  |       |
| <b>12</b> | <b>VENDRE DES SITES 3D</b>                          | p. 37 |
|           | À qui, comment présenter, quel prix                 |       |
| <b>+</b>  | <b>ANNEXES</b>                                      | p. 40 |
|           | Checklist, prompts, glossaire, code complet         |       |

A 3D rendered scene of a tropical cafe. The cafe has a thatched roof and is surrounded by palm trees. Inside, there is a counter with a coffee machine and various items. The scene is dimly lit, with some lights hanging from the ceiling.

# 01

## CE QU'EST UN SITE 3D CINÉMATIQUE

Un site qu'on ne parcourt pas : on y entre. Chaque mouvement de la molette fait avancer la caméra dans le lieu.

# LE SCROLL DEVIENT UNE CAMÉRA

Sur un site classique, on fait défiler des blocs de texte et des photos. Sur un site 3D cinématique, le scroll pilote une caméra qui avance dans une scène continue.



Mon film d'ouverture : du couloir au bureau, dans l'écran, puis dans le café jusqu'à la tasse. Un seul mouvement de caméra, piloté par le scroll.

Techniquement, c'est une **vidéo** dont on choisit l'image affichée en fonction de la position du scroll. Quand le visiteur descend de 10 %, la vidéo avance de 10 %. Quand il remonte, elle recule. Il a l'impression de conduire la caméra lui-même.

Cette vidéo n'est pas filmée. Elle est fabriquée à partir d'une série d'images qui s'emboîtent : chaque image est un détail de la précédente. En zoomant progressivement de l'une à l'autre, on obtient un travelling continu, sans coupure, qui donne une vraie sensation de profondeur. C'est ce qu'on appelle ici la « 3D » : une profondeur ressentie, pas un modèle 3D.

## À RETENIR

**Images qui s'emboîtent** → **film à zoom continu** → **vidéo pilotée par le scroll**. Tout le guide suit ces trois étapes.

# TROIS RAISONS POUR LESQUELLES LES GENS RESTENT

## 01

### LA CURIOSITÉ

Le visiteur veut voir la suite. Chaque scroll révèle un nouveau lieu : il descend jusqu'au bout pour savoir où la caméra l'emmène.

## 02

### LA MÉMOIRE

On se souvient d'un voyage, pas d'une grille de photos. Le lieu est raconté dans l'ordre où on le découvre en vrai.

## 03

### LA DIFFÉRENCE

Presque aucun commerce local n'a ça. Le site se partage, se montre, se commente : il devient un argument en soi.

#### ► POUR QUELS LIEUX ?

L'effet donne le meilleur de lui-même quand il y a un **chemin** à raconter : on arrive, on entre, on s'installe, on découvre le produit.

| TYPE DE LIEU          | LE VOYAGE DE LA CAMÉRA                                                    |
|-----------------------|---------------------------------------------------------------------------|
| Café, restaurant      | La rue ou la plage, la façade, la salle, le comptoir, le plat ou la tasse |
| Hôtel, maison d'hôtes | Le paysage, l'entrée, le salon, la chambre, la vue depuis le lit          |
| Boutique, marque      | L'origine du produit, l'atelier, les mains qui fabriquent, l'objet fini   |
| Artisan               | Le chantier, l'outil, le geste, le détail de la finition                  |

#### CE QUE CE N'EST PAS

Ce n'est pas un jeu vidéo en 3D temps réel, et ce n'est pas un logiciel de modélisation. Pas besoin de Blender ni de carte graphique puissante : tout repose sur des images et une vidéo. C'est ce qui rend la méthode rapide, légère, et accessible sans être développeur.



# 02

## LE PRINCIPE DE LA POUPÉE RUSSE

Une seule règle, mais elle décide de tout : chaque image doit contenir la suivante.

# CHAQUE PLAN CONTIENT LE SUIVANT

Imagine des poupées russes. La plus grande contient la moyenne, qui contient la petite. Ton film fonctionne pareil : le plan 2 est une zone du plan 1, le plan 3 une zone du plan 2.



La caméra part du plan 1 et zoome vers la zone où se trouve le plan 2. Juste avant d'arriver, le plan 2, plus détaillé, prend le relais en fondu. On continue à zoomer, et ainsi de suite. Le spectateur ne voit jamais de coupure : il voit une caméra qui avance.

## ► LES TROIS CONDITIONS

1. **Même point de vue.** Le plan suivant est pris dans la direction où la caméra avance. Pas de changement d'angle brutal.
2. **Même lumière, même heure, même météo.** Un coucher de soleil ne peut pas devenir un plein midi au plan suivant.
3. **Un zoom raisonnable.** Entre deux plans, le détail doit occuper entre 60 % et 80 % de la largeur du plan précédent. Au-delà, le saut se voit.

### LE CHIFFRE À RETENIR

Mes meilleurs raccords ont une échelle de **0,60 à 0,80**. Mon script te l'affiche pour chaque raccord : si tu vois 0,40, il manque un plan intermédiaire (chapitre 04).

# UNE VRAIE SÉQUENCE, DÉCORTIQUÉE



Six plans de mon site Mộc Sương, dans l'ordre. Chaque image est le centre agrandi et enrichi de la précédente.

Regarde bien : le pavillon au fond du plan 1 devient le sujet du plan 2, le chemin de pierre du plan 2 mène à la porte du plan 3. Rien n'est laissé au hasard. Le sujet du plan suivant est déjà **visible, au centre**, dans le plan actuel.



Le même début monté par mon script : 12 images prises toutes les 10 images du film. La caméra avance sans coupure.



# 03

## ÉCRIRE LE STORYBOARD

Avant de générer la moindre image, écris le voyage. Dix minutes de storyboard t'évitent deux heures de plans ratés.

# DU PLUS LARGE AU PLUS DÉSIRABLE

Un bon voyage commence loin et finit sur ce que le client est venu chercher : la tasse, le plat, la chambre, l'objet.

## 01 LE PLAN D'OUVERTURE

Large, beau, reconnaissable. Il pose le décor : la plage, la vallée, la rue. C'est aussi l'image qui s'affiche avant le chargement de la vidéo, donc elle doit être forte toute seule.

## 02 L'APPROCHE

Deux à quatre plans pour se rapprocher : la façade, l'entrée, le seuil. C'est là que le visiteur comprend qu'il va entrer.

## 03 L'INTÉRIEUR

Deux à quatre plans dans le lieu : la salle, le comptoir, l'atelier, les gestes. C'est le cœur de l'identité.

## 04 LA SCÈNE FINALE

Le produit, en gros plan, parfait. La tasse fumante, le plat, le lit défait face à la mer. C'est l'image qui donne envie de venir, et c'est là que tu places ton bouton d'appel.

### COMBIEN DE PLANS ?

**6 à 8 plans** pour un premier site. **12 à 14** pour un rendu très fluide, en ajoutant des plans intermédiaires. Mon film d'ouverture en compte 13, pour 36 secondes de voyage.

# LE STORYBOARD D'UN CAFÉ DE PLAGE

Voici le storyboard réel de ma démo Kafe Talay. Recopie ce tableau pour ton projet : une ligne par plan, le sujet au centre, et ce qui doit être déjà visible pour préparer le plan suivant.

| PLAN | CE QU'ON VOIT                                             | DÉJÀ VISIBLE AU CENTRE |
|------|-----------------------------------------------------------|------------------------|
| 01   | La plage au coucher du soleil, cocotiers, le café au loin | Le toit du café        |
| 02   | La façade en bois, le chemin de planches dans le sable    | L'entrée sous le toit  |
| 03   | Sous le toit, les tables, les lampes allumées             | Le comptoir            |
| 04   | Le comptoir, la machine à café, les bocaux                | Une tasse posée        |
| 05   | La tasse, latte art, grains de café, sachet kraft         | Fin du voyage          |

## ► LES TITRES QUI ACCOMPAGNENT LE VOYAGE

Chaque étape du voyage reçoit un titre court qui apparaît pendant le scroll. Écris-les en même temps que le storyboard :

| MOMENT        | TITRE                     | RÔLE                            |
|---------------|---------------------------|---------------------------------|
| Plan 01       | « Approchez. »            | Promesse, invitation à scroller |
| Plans 02 à 03 | « Entrez. »               | L'expérience du lieu            |
| Plan 04       | « Installez-vous. »       | Le moment vécu                  |
| Plan 05       | « Réservez votre table. » | L'appel à l'action              |

A person wearing a traditional conical hat is harvesting tea in a misty mountain landscape. The person is crouched in a tea field, with a large woven basket on their back. The background shows rolling hills and mountains shrouded in mist.

04

# GÉNÉRER LES PLANS AVEC L'IA

Le secret n'est pas le prompt parfait. C'est de toujours partir de l'image précédente.

# TOUJOURS PARTIR DU PLAN PRÉCÉDENT

Si tu génères chaque plan de zéro, tu obtiens de belles images qui ne se raccordent pas. La cohérence vient d'une seule habitude : donner le plan précédent comme image de référence.

## 01 GÉNÈRE LE PLAN D'OUVERTURE

Format paysage 16:9, le plus grand possible. Décrit le lieu, l'heure, la lumière, le style photo. Garde ce prompt : tu vas réutiliser ses mots dans tous les plans suivants.

```
# Prompt du plan 01
Photo réaliste, format paysage 16:9. Un petit café de plage en bois
au bord de l'eau, au coucher du soleil, cocotiers, sable blanc,
lumière chaude et dorée venant de la droite. Vue large depuis la
plage, le café au centre de l'image, à mi-distance. Style photo de
voyage haut de gamme, pas de texte, pas de logo, pas de personnes.
```

## 02 GÉNÈRE CHAQUE PLAN SUIVANT DEPUIS LE PRÉCÉDENT

Fournis l'image précédente et demande explicitement d'avancer vers le centre, en gardant tout le reste identique.

```
# Prompt des plans suivants (avec l'image précédente jointe)
Voici le plan précédent de mon film. Génère le plan suivant :
la caméra a avancé tout droit vers le centre de l'image, environ
1,4 fois plus près. On voit maintenant [la façade du café, le chemin
de planches]. Garde exactement la même lumière, la même heure, les
mêmes couleurs, le même style photo, format 16:9. Pas de texte.
```

### L'ASTUCE QUI CHANGE TOUT

Mentionne toujours « la caméra a avancé tout droit vers le centre ». Si l'outil déplace le sujet sur le côté, le raccord devient difficile : régénère plutôt que de garder une image décentrée.

# QUAND LE SAUT EST TROP GRAND

Entre deux plans très différents, la plage et la tasse par exemple, le zoom est trop fort : le fondu se voit. La solution est un plan intermédiaire, fabriqué avec ma technique du recadrage à 79 %.

1. **Recadre le centre du plan N** à 79 % de sa largeur et de sa hauteur, puis agrandis ce recadrage au format d'origine. Tu obtiens une image un peu floue mais parfaitement placée : c'est ta **référence géométrique**.
2. **Donne à l'IA cette référence et le plan N+1**, et demande une image exactement à mi-chemin : le cadrage de la référence, avec les détails nets et fidèles.
3. **Nomme le plan N-bis** (01a, 02a...) et place-le entre les deux.

```
# Recadrage à 79 % avec ffmpeg (une ligne)
ffmpeg -i plan-02.png -vf "crop=iw*0.79:ih*0.79,scale=1672:941" reference-02a.png
```

```
# Prompt du plan intermédiaire (joindre reference-02a.png et plan-03.png)
Image 1 : cadrage exact à respecter (recadrage du plan précédent).
Image 2 : le plan vers lequel la caméra avance.
Génère une photo nette avec exactement le cadrage de l'image 1,
en ajoutant les détails réalistes qu'on découvrirait en se
rapprochant, fidèles à l'image 2. Même lumière, format 16:9.
```

## ► CONTRÔLE QUALITÉ DE CHAQUE PLAN

- ☐ Le sujet du plan suivant est visible et centré
- ☐ Aucun texte inventé, aucun logo de marque réelle
- ☐ Mains et visages sans défaut (cinq doigts, regard naturel)
- ☐ Même heure et même couleur de lumière que le plan précédent

# UN DOSSIER PROPRE, UN FILM PROPRE

Mon script lit les images dans l'ordre alphabétique. Adopte dès le départ une numérotation qui ne casse jamais l'ordre :

```
mon-projet/  
  plans/  
    01.png    la plage  
    01a.png   intermédiaire ajouté plus tard  
    02.png    la façade  
    03.png    sous le toit  
    04.png    le comptoir  
    05.png    la tasse  
  hd/         les versions agrandies (chapitre 06)  
  site/       la page et les vidéos (chapitres 08 à 10)
```

« 01a » se range naturellement entre « 01 » et « 02 ». Tu peux donc ajouter un plan intermédiaire à tout moment sans renommer le reste.

## FORMAT RECOMMANDÉ

PNG, **16:9**, au moins **1672 × 941** pixels. Mon script recadre automatiquement si une image n'est pas exactement au bon format, mais il vaut mieux générer directement en 16:9.

## GARDE TES PROMPTS

Note chaque prompt dans un fichier texte à côté des images. Quand un client demande une retouche six mois plus tard, tu régénères en deux minutes au lieu de tout réinventer.

# 05

## TRAVAILLER AVEC DE VRAIES PHOTOS

Un site pour un vrai commerce doit ressembler au vrai commerce. L'IA sublime, elle n'invente pas.

# SUBLIMER SANS MENTIR

Le client reçoit des visiteurs qui ont vu le site. S'ils trouvent une salle plus petite ou un plat différent, c'est la confiance qui part. La règle : on améliore la lumière et le cadrage, on n'ajoute rien qui n'existe pas.

## 01 PARTIR DES PHOTOS DU CLIENT

Demande quatre photos : l'extérieur, l'intérieur, un plat ou une boisson, la vue. Utilise-les comme images de référence pour chaque plan correspondant, avec cette consigne :

Voici une vraie photo du lieu. Génère une version plus lumineuse, mieux cadrée, au format 16:9, en gardant exactement les mêmes meubles, couleurs, matériaux et objets. N'ajoute aucun élément qui n'est pas sur la photo.

## 02 LES PERSONNES

Si une personne apparaît dans le film (le patron, toi-même), deux règles :

- ▶ **Son accord écrit** avant toute publication.
- ▶ **La ressemblance se travaille en retouche locale.** Génère d'abord la scène, puis retouche uniquement le visage en fournissant deux ou trois vraies photos de la personne comme référence. Un visage généré « au hasard » est lisse et générique : les proches ne s'y reconnaissent pas.

### MON EXPÉRIENCE

Sur mon propre film, le premier visage généré ne me ressemblait pas. C'est la retouche du seul visage, avec mes photos en référence, qui a tout changé : barbe, forme des yeux, mâchoire. Tout le reste de l'image est resté identique au pixel près.

# CORRIGER UN DÉTAIL SANS TOUT CASSER

Un plan est presque parfait, mais un détail cloche : une télé à la place d'un écran d'ordinateur, une main à six doigts, un logo inventé. Ne régénère pas tout le plan, tu perdrais le raccord avec les autres. Fais une retouche locale.

1. **Recadre la zone à corriger** avec une bonne marge autour.
2. **Retouche ce recadrage** avec l'IA, en décrivant uniquement la correction.
3. **Recolle la zone** dans l'image d'origine avec un contour adouci.
4. **Vérifie** que rien n'a bougé en dehors de la zone : c'est ce qui garantit que le raccord avec les plans voisins reste parfait.

```
# Prompt de retouche locale
Retouche uniquement [l'écran posé sur le bureau] : remplace-le par
un moniteur d'ordinateur à fin cadre noir qui affiche un site de
café de plage. Tout le reste de l'image doit rester strictement
identique : lumière, perspective, objets, personnes.
```

## ► ANIMER UN GESTE

Même principe pour faire bouger une main : génère deux ou trois variantes où **seule la main change** (ouverte, à moitié fermée, fermée), puis alterne-les pendant le film. Deux gestes lents valent mieux que vingt rapides : un geste trop répété paraît mécanique.

A dark, atmospheric landscape with misty mountains and a tea set in the foreground. The scene is dimly lit, with the mountains in the background appearing as soft, hazy shapes. In the foreground, a dark wooden table holds a traditional tea set, including a teapot and two cups, which are slightly out of focus. The overall mood is serene and contemplative.

# 06

## LA HAUTE DÉFINITION

Quand la caméra zoome, chaque pixel est agrandi. Une image moyenne devient floue au bout de trois plans.

# DOUBLER LA DÉFINITION DE CHAQUE PLAN

Pendant un raccord, la caméra affiche parfois seulement 60 % de l'image. Sur un écran large, ces 60 % sont agrandis : c'est là que le flou apparaît. La parade est de travailler avec des plans deux fois plus grands que l'écran final.

## 01 INSTALLER REAL-ESRGAN (GRATUIT)

Télécharge la version de ton système (Mac, Windows ou Linux) sur la page des versions du projet Real-ESRGAN sur GitHub (fichier `realesrgan-ncnn-vulkan`), puis décompresse-la. Aucune installation, aucun compte.

## 02 AGRANDIR UN PLAN

```
# Agrandissement x4 avec le modèle photo réaliste
./realesrgan-ncnn-vulkan -i plans/02.png -o hd/02.png -n realesrgan-x4plus
```

## 03 REVENIR À X2 AVEC UN RENDU NATUREL

Le x4 brut donne parfois un aspect « plastique ». Ma recette : réduire le x4 à deux fois la taille d'origine, puis le mélanger à 60 % avec un agrandissement classique, et ajouter un grain photo très léger. Le résultat est net sans être artificiel.

```
# Ramener le x4 à x2 (exemple pour un plan de 1672 × 941)
ffmpeg -i hd/02.png -vf "scale=3344:1882:flags=area" hd/02-x2.png
```

### POURQUOI PAS DIRECTEMENT X4 ?

Un fichier x4 de 6688 pixels de large alourdit tout le montage pour un gain invisible. Le x2 (environ 3300 pixels) suffit pour un rendu parfait en 1080p et même au-delà.

# CE QUE L'AGRANDISSEMENT NE FAIT PAS

L'agrandissement rend l'image plus nette. Il ne la rend pas plus fidèle. Si un visage fait 50 pixels dans le plan d'origine, l'agrandir donnera un visage net... mais toujours approximatif.

## CE QUE FAIT L'AGRANDISSEMENT

- ▶ Des contours nets
- ▶ Des textures crédibles (bois, feuillage, tissu)
- ▶ Un film qui reste propre en plein écran

## CE QU'IL NE FAIT PAS

- ▶ Rendre un visage ressemblant
- ▶ Corriger une main mal dessinée
- ▶ Réparer un texte illisible

### L'ordre des opérations compte :

1. Plans générés et validés
2. Agrandissement x2
3. Retouches locales sur les versions agrandies (visage, mains), avec les photos de référence
4. Montage du film

Retoucher après l'agrandissement permet de travailler le visage à haute définition : l'IA a plus de pixels pour dessiner une barbe de trois jours, un regard, une vraie peau.



# 07

## **MONTER LE FILM**

Mon script `film_zoom.py` assemble tes plans en un seul travelling continu. Une commande.

# PRÉPARER TON ORDINATEUR

À faire une seule fois. Compte dix minutes.

## 01 PYTHON 3

Sur Mac, Python 3 est souvent déjà là : tape `python3 --version` dans le Terminal. Sinon, télécharge-le sur [python.org](https://python.org). Sur Windows, coche « Add Python to PATH » pendant l'installation.

## 02 FFMPEG

L'outil qui fabrique les vidéos. Sur Mac avec Homebrew : `brew install ffmpeg`. Sur Windows : télécharge une version sur [ffmpeg.org](https://ffmpeg.org) et ajoute son dossier au PATH.

## 03 LES DEUX BIBLIOTHÈQUES DU SCRIPT

```
pip install opencv-python numpy
```

## 04 VÉRIFIER

```
python3 -c "import cv2, numpy; print('OK')"  
ffmpeg -version
```

Si les deux commandes répondent sans erreur, tu es prêt.

### OÙ TROUVER LES OUTILS

Les trois fichiers ( `film_zoom.py` , `encoder_web.sh` , `page_scroll.html` ) sont fournis avec ce guide dans le dossier « outils ». Leur code complet est aussi imprimé en annexe, pour que tu ne dépendes de rien.

# UNE COMMANDE, UN FILM

```
python3 film_zoom.py hd film.mp4
```

Le script lit toutes les images du dossier `hd` dans l'ordre, calcule pour chaque raccord où se trouve le plan suivant dans le plan actuel, puis fabrique le film. Il t'affiche un compte rendu :

```
01.webp → 02.webp : échelle 0.61, centre (0.49, 0.51), calé automatiquement
02.webp → 03.webp : échelle 0.70, centre (0.50, 0.51), calé automatiquement
03.webp → 04.webp : échelle 0.64, centre (0.49, 0.46), calé automatiquement
Film prêt : film.mp4
```

## ► LIRE LE COMPTE RENDU

| TU VOIS                                           | ÇA VEUT DIRE                                             | QUE FAIRE                                                               |
|---------------------------------------------------|----------------------------------------------------------|-------------------------------------------------------------------------|
| <b>calé automatiquement</b> , échelle 0,55 à 0,85 | Le script a retrouvé le plan suivant dans le plan actuel | Rien, c'est parfait                                                     |
| <b>zoom central par défaut</b>                    | Les deux plans sont trop différents pour être calés      | Normal pour un changement de pièce. Sinon, ajoute un plan intermédiaire |
| échelle <b>inférieure à 0,5</b>                   | Le saut est trop grand                                   | Ajoute un plan intermédiaire (chapitre 04)                              |

## ► LES RÉGLAGES UTILES

| OPTION                                     | EFFET                                                                   |
|--------------------------------------------|-------------------------------------------------------------------------|
| <code>--images-par-raccord 90</code>       | Durée de chaque raccord. 90 images = 3 secondes à 30 images par seconde |
| <code>--zoom 0.78</code>                   | Zoom utilisé quand le calage automatique échoue                         |
| <code>--largeur 1920 --hauteur 1080</code> | Taille du film. Garde 16:9                                              |

# COMMENT LE SCRIPT FABRIQUE L'ILLUSION

Tu n'as pas besoin de comprendre ces détails pour t'en servir, mais ils t'aideront à diagnostiquer un raccord qui ne te plaît pas.

## 01 LE CALAGE

Le script repère des milliers de points caractéristiques (angles, textures) dans les deux plans et cherche comment superposer le plan suivant dans le plan actuel. Il en déduit une **échelle** et un **centre**.

## 02 LE ZOOM EXPONENTIEL

La caméra ne zoome pas à vitesse constante en pixels, mais en proportion :  $\text{zoom} = \text{échelle} ^ t$ . C'est ce qui donne une sensation d'avancée régulière, comme un vrai travelling, au lieu d'un zoom qui accélère.

## 03 LE FONDU POSÉ

Dans la seconde moitié du raccord, le plan suivant apparaît progressivement, posé exactement à sa place et avec des bords adoucis. Quand la caméra arrive, il occupe tout l'écran : on est passé d'un plan à l'autre sans coupure visible.

## 04 LA PAUSE FINALE

Le film se termine par une seconde de pause sur le dernier plan : le visiteur a le temps de voir la scène finale avant le texte et le bouton d'appel.

### CONTRÔLER LA FLUIDITÉ

Ouvre le film et fais défiler la barre de lecture lentement. Si tu vois un « saut » à un raccord, c'est presque toujours un plan décentré : régénère-le avec le sujet au centre, ou ajoute un intermédiaire.



# 08

## ENCODER POUR LE WEB

Un film magnifique qui rame au scroll ne sert à rien. L'encodage fait toute la différence.

# FLUIDE AU SCROLL, LÉGER AU CHARGEMENT

Une vidéo normale est optimisée pour être lue du début à la fin. Toi, tu veux que le navigateur saute n'importe où, instantanément, dans les deux sens. Ça se règle à l'encodage.

```
bash encoder_web.sh film.mp4
```

Le script produit quatre fichiers : la vidéo ordinateur (1920 × 1080), la vidéo téléphone (1080 × 1920, verticale), et leurs deux images d'attente.

## ► CE QUI COMPTE DANS LA COMMANDE

| RÉGLAGE                           | POURQUOI                                                                                                                                                   |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-g 8</code>                 | <b>Une image clé toutes les 8 images.</b> Le navigateur peut se positionner partout sans décoder de longues séquences. C'est le réglage le plus important. |
| <code>-bf 0</code>                | Pas d'images « bidirectionnelles » : le retour en arrière reste instantané.                                                                                |
| <code>-crf 26</code>              | Le compromis qualité et poids. 22 = plus net et plus lourd, 30 = plus léger et plus doux.                                                                  |
| <code>-movflags +faststart</code> | La vidéo commence à jouer avant d'être entièrement téléchargée.                                                                                            |
| <code>-an</code>                  | Pas de son : une vidéo muette peut démarrer seule sur téléphone.                                                                                           |

### MON REPÈRE DE POIDS

Pour 36 secondes de voyage en 1080p : environ **25 Mo** sur ordinateur et **20 Mo** sur téléphone avec une qualité très haute. Si ton film est plus long, monte le `crf` de 2 points plutôt que de réduire la définition.

# ENCODER\_WEB.SH

```
#!/bin/bash
# encoder_web.sh · prépare ton film pour le web (défilement fluide au scroll)
# Utilisation : bash encoder_web.sh film.mp4
# Crée film-web.mp4 (ordinateur, 1920x1080) et film-mobile.mp4 (téléphone, 1080x1920)
set -e
IN="$1"; BASE="${IN%.*}"
# Une image clé toutes les 8 images : le navigateur peut sauter partout dans la vidéo sans
# ramer.
ffmpeg -y -i "$IN" -an -vf "scale=1920:1080:flags=lanczos" -c:v libx264 -preset slow -tune
film -crf 26 -g 8 -bf 0 -pix_fmt yuv420p -movflags +faststart "$BASE-web.mp4"
# Version téléphone : on garde le centre de l'image au format vertical.
ffmpeg -y -i "$IN" -an -vf "crop=ih*9/16:ih,scale=1080:1920:flags=lanczos" -c:v libx264 -
preset slow -tune film -crf 26 -g 8 -bf 0 -pix_fmt yuv420p -movflags +faststart "$BASE-
mobile.mp4"
# Image d'attente affichée avant le chargement de la vidéo.
ffmpeg -y -i "$BASE-web.mp4" -frames:v 1 -q:v 3 "$BASE-poster.jpg"
ffmpeg -y -i "$BASE-mobile.mp4" -frames:v 1 -q:v 3 "$BASE-poster-mobile.jpg"
echo "Prêt : $BASE-web.mp4, $BASE-mobile.mp4 et les deux images d'attente."
```

## ► LA VERSION TÉLÉPHONE

Sur un téléphone tenu en portrait, un film 16:9 serait réduit à une petite bande. Le script découpe donc le centre de l'image au format vertical. Si ton sujet n'est pas au centre dans les premiers plans (une personne sur la gauche, par exemple), ajuste la position du découpage :

```
# Découper à 34 % depuis la gauche au lieu du centre
-vf "crop=ih*9/16:ih:(iw-ih*9/16)*0.34:0,scale=1080:1920"
```

Pour mon film d'ouverture, le découpage suit la personne à gauche pendant le premier tiers, puis revient progressivement au centre quand la caméra entre dans l'écran. C'est un réglage avancé, mais c'est ce détail qui rend la version téléphone aussi belle que la version ordinateur.



09

# LA PAGE QUI FAIT DÉFILER LE FILM

Soixante lignes de code, et le scroll devient une caméra.

# PAGE\_SCROLL.HTML, EN TROIS IDÉES

Place `page_scroll.html` (renommé `index.html`) dans le même dossier que tes vidéos et tes images d'attente. Ouvre-le : ton site fonctionne déjà.

## 01 UNE SCÈNE COLLÉE À L'ÉCRAN

Le bloc `.film` est très haut ( `400vh` , soit quatre écrans). À l'intérieur, la scène est « collée » en haut de l'écran ( `position: sticky` ) pendant qu'on fait défiler ces quatre écrans. La hauteur règle la durée du voyage : plus elle est grande, plus il faut scroller.

## 02 LE SCROLL DEVIENT UN TEMPS DE VIDÉO

```
function avancement(){
  var h = film.offsetHeight - innerHeight;
  return Math.min(1, Math.max(0, -film.getBoundingClientRect().top / h));
}
cible = avancement() * (video.duration - .05);
```

À 0 % de scroll, on affiche le début de la vidéo, à 100 % la fin. Entre les deux, tout est proportionnel.

## 03 LA CAMÉRA RATTRAPE LE SCROLL EN DOUCEUR

```
if (Math.abs(cible - actuel) > .008) {
  actuel += (cible - actuel) * .2;
  video.currentTime = actuel;
}
```

Au lieu de sauter directement à la bonne image, la vidéo parcourt 20 % de la distance restante à chaque rafraîchissement d'écran. Ce petit amorti donne la sensation d'un vrai mouvement de caméra, même avec une molette saccadée.

# LES TITRES, LE TÉLÉPHONE, L'IPHONE

## ► LES TITRES QUI APPARAISSENT

```
<div class="titre" data-de=".3" data-a=".6"><h2>Entrez</h2></div>
```

Chaque titre a une fenêtre d'apparition : `data-de` et `data-a`, entre 0 (début du voyage) et 1 (fin). Place-les en regardant ton film : à quel pourcentage arrive-t-on au comptoir ? Mets le titre juste avant.

## ► ORDINATEUR OU TÉLÉPHONE

```
var mobile = innerWidth / innerHeight < .8;
video.src = mobile ? 'film-mobile.mp4' : 'film-web.mp4';
```

Un écran plus haut que large reçoit la version verticale. Chacun télécharge une seule vidéo : la page reste légère.

## ► LE DÉTAIL IPHONE

Safari sur iPhone refuse de positionner une vidéo tant qu'elle n'a jamais été lue. Le modèle lance donc une lecture de quelques millisecondes au premier toucher de l'écran, puis la met en pause. Sans ce détail, le film reste figé sur iPhone.

### LA SUITE DE LA PAGE

Après le film, c'est une page normale : présentation, carte ou menu, horaires, plan, contact. Garde un **bouton d'appel visible dès la scène finale** du film : c'est le moment où l'envie est la plus forte.

# EN LIGNE EN CINQ MINUTES

Netlify héberge gratuitement les sites comme le tien, avec une adresse sécurisée en https. Pas de serveur à configurer.

1. Rassemble dans un dossier `site` : `index.html` , `film-web.mp4` , `film-mobile.mp4` , `film-poster.jpg` , `film-poster-mobile.jpg` .
2. Crée un compte gratuit sur netlify.com.
3. Ouvre la page « Netlify Drop » et **glisse-dépose le dossier** `site` .
4. Ton site est en ligne à une adresse du type `nom-au-hasard.netlify.app` . Tu peux la renommer dans les réglages du site.

## ► UN VRAI NOM DE DOMAINE

Pour un client, achète son domaine (environ 10 € par an chez un registraire comme OVH ou Gandi), puis ajoute-le dans Netlify, rubrique « Domain management ». Netlify t'indique les deux réglages DNS à faire chez le registraire et crée le certificat https automatiquement.

## ► VÉRIFIER AVANT D'ENVOYER AU CLIENT

- ☐ Sur ordinateur : le film suit le scroll dans les deux sens
- ☐ Sur un vrai iPhone et un vrai Android, pas seulement en simulation
- ☐ En 4G, pas seulement en Wi-Fi : l'image d'attente s'affiche tout de suite
- ☐ Les titres apparaissent au bon moment
- ☐ Le bouton d'appel fonctionne (téléphone, WhatsApp ou formulaire)



# 11

# LES PIÈGES QUE J'AI PAYÉS

Huit erreurs que j'ai faites sur de vrais projets, pour que tu ne les fasses pas.

# HUIT ERREURS, HUIT PARADES

| LE PROBLÈME                                                                               | LA PARADE                                                                                                                                                      |
|-------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Les lignes droites qui ondulent</b><br>(poteaux, cadres de fenêtre) pendant un raccord | Arrive quand on force une déformation entre deux plans trop différents. Mon script fait un zoom et un fondu propres, sans déformation : garde-le tel quel.     |
| <b>Le fantôme</b> : une personne à moitié transparente pendant un fondu                   | Deux plans où la personne change de position (debout puis assise). Raccourcis ce raccord ou fais disparaître la personne un plan plus tôt.                     |
| <b>Le flou en plein écran</b>                                                             | La vidéo a été réduite pour gagner du poids. Encode en 1920 de large minimum et joue sur le <code>crf</code> , pas sur la définition.                          |
| <b>Le film qui rame au scroll</b>                                                         | Vidéo encodée sans <code>-g 8 -bf 0</code> . Réencode avec <code>encoder_web.sh</code> .                                                                       |
| <b>Le visage générique</b>                                                                | Retouche locale du seul visage, sur la version agrandie, avec de vraies photos en référence (chapitres 05 et 06).                                              |
| <b>Le texte inventé</b> sur une enseigne ou un écran                                      | Précise « pas de texte lisible » dans chaque prompt, et vérifie chaque plan en plein écran avant le montage.                                                   |
| <b>La personne coupée sur téléphone</b>                                                   | Le découpage vertical prend le centre. Décale-le vers ton sujet (chapitre 08).                                                                                 |
| <b>Le changement d'un plan qui décale tout</b>                                            | Si tu remplaces un plan, vérifie la durée des raccords voisins : un raccord mieux calé peut allonger ou raccourcir une partie du voyage et décaler tes titres. |

# REGARDE TON FILM IMAGE PAR IMAGE AVANT DE LE MONTRER.

Presque tous les défauts que tu viens de lire se voient en trois secondes quand on avance la vidéo lentement, et presque jamais quand on la lit à vitesse normale. Les clients, eux, scrollent lentement : ils verront.

## ► MA ROUTINE DE CONTRÔLE

1. **Planche contact** : une image toutes les secondes, côte à côte. Les incohérences de lumière sautent aux yeux.
2. **Lecture lente** de chaque raccord, en avançant image par image.
3. **Plein écran sur un grand moniteur** pour le flou.
4. **Vrai téléphone** pour le cadrage vertical et la fluidité.
5. **Un regard extérieur** : montre-le à quelqu'un sans rien expliquer et observe où il fronce les sourcils.

```
# Planche contact : une image par seconde, 6 colonnes  
ffmpeg -i film.mp4 -vf "fps=1,scale=320:-1,tile=6x6" -frames:v 1 planche.jpg
```

A black and white photograph of a woman in a field, wearing a headscarf and a button-down shirt, carefully sorting tea leaves into a large woven basket. The background shows a vast field of tea plants under a hazy sky. In the bottom right corner, there is a small image of a tea canister labeled 'Mộc Sương GREEN TEA'.

# 12

## VENDRE DES SITES 3D

Tu sais les fabriquer. Voici comment les proposer à ceux qui en ont besoin.

# MONTRE AVANT DE VENDRE

Un site 3D cinématique se vend en dix secondes de vidéo, pas en dix lignes de description. Ta meilleure arme est une démo qui bouge.



Trois de mes démos : une marque de thé au Vietnam, un café de montagne en Thaïlande, un restaurant de kebab. Trois univers, la même méthode.

## 01 CHOISIS TES CIBLES

Les commerces où l'**ambiance** fait vendre : cafés, restaurants, hôtels, maisons d'hôtes, boutiques de créateurs, lieux touristiques. Évite au début les activités où le client cherche surtout un prix ou un numéro de téléphone.

## 02 PRÉPARE UNE DÉMO À LEUR IMAGE

Un storyboard de cinq plans avec les photos publiques du commerce, un film court, une page en ligne. Tu arrives avec **leur** lieu en mouvement : c'est incomparable avec une présentation générique.

## 03 MONTRE LA VIDÉO DE TA MAIN QUI SCROLLE

Filme ton écran pendant que tu fais défiler le site. Cette vidéo sert partout : message privé, réseaux sociaux, rendez-vous. C'est elle qui déclenche le « je veux la même chose ».

# FIXER TON PRIX, REMPLIR TON AGENDA

## ► QUEL PRIX ?

Un site 3D cinématique demande plus de travail qu'un site vitrine classique, et il se remarque beaucoup plus. À titre de repère, mon offre de site cinématique est à **590 € en paiement unique**, sans abonnement. Adapte à ton marché, à ton expérience et à ce que tu inclus (nombre de plans, langues, retouches, nom de domaine).

### CE QUE J'INCLUS DANS UNE OFFRE

- Le voyage en film à partir des photos du lieu
- Les versions ordinateur et téléphone
- Menu ou carte, horaires, plan, contact
- Deux langues si la clientèle est internationale
- La mise en ligne et le branchement du nom de domaine

## ► LE TUNNEL QUI TRANSFORME UNE VIDÉO EN RENDEZ-VOUS

1. **La vidéo** montre ta main qui scrolle un site, avec à l'écran : « Commente 3D ».
2. **Le commentaire** déclenche un message privé automatique (avec un outil comme ManyChat).
3. **Le message** vérifie que la personne te suit, puis envoie ta page.
4. **La page** propose deux choix : réserver un appel pour faire créer son site, ou apprendre à le faire.

C'est exactement le chemin qui t'a amené jusqu'à ce guide.

# LA CHECKLIST COMPLÈTE

## 01 PRÉPARER

- ☐ Storyboard écrit, du large au désirable
- ☐ Sujet de chaque plan visible au centre du précédent
- ☐ Titres du voyage écrits
- ☐ Photos du client récupérées (4 minimum)
- ☐ Accord écrit des personnes filmées

## 02 GÉNÉRER

- ☐ Plan 01 en 16:9, grand format
- ☐ Chaque plan généré depuis le précédent
- ☐ Même lumière et même heure partout
- ☐ Aucun texte inventé, aucun logo
- ☐ Mains et visages vérifiés
- ☐ Plans intermédiaires là où le saut est grand

## 03 AGRANDIR ET RETOUCHER

- ☐ Tous les plans agrandis en x2
- ☐ Visages retouchés avec les photos de référence
- ☐ Rien n'a bougé hors des zones retouchées

## 04 MONTER

- ☐ film\_zoom.py lancé, compte rendu lu
- ☐ Échelles entre 0,55 et 0,85
- ☐ Film vérifié image par image
- ☐ Planche contact sans rupture de lumière

## 05 PUBLIER

- ☐ encoder\_web.sh lancé
- ☐ Découpage téléphone vérifié
- ☐ Titres calés sur le film
- ☐ Bouton d'appel sur la scène finale
- ☐ Testé sur vrai iPhone et vrai Android
- ☐ Testé en 4G
- ☐ Nom de domaine branché, https actif

## 06 VENDRE

- ☐ Vidéo de ta main qui scrolle
- ☐ Démo au nom du prospect
- ☐ Offre claire, prix affiché

# LES PROMPTS PRÊTS À COPIER

## 01 PLAN D'OUVERTURE

Photo réaliste, format paysage 16:9. [Le lieu] vu de loin, [l'heure et la lumière]. Le sujet au centre de l'image, à mi-distance. Style photo [de voyage / d'architecture / culinaire] haut de gamme. Pas de texte, pas de logo, pas de personnes.

## 02 PLAN SUIVANT

Voici le plan précédent de mon film. Génère le plan suivant : la caméra a avancé tout droit vers le centre, environ 1,4 fois plus près. On voit maintenant [nouveau sujet]. Même lumière, même heure, mêmes couleurs, même style, format 16:9. Pas de texte.

## 03 PLAN INTERMÉDIAIRE

Image 1 : cadrage exact à respecter. Image 2 : le plan vers lequel la caméra avance. Génère une photo nette avec le cadrage de l'image 1 et les détails fidèles à l'image 2. Même lumière, format 16:9.

## 04 À PARTIR D'UNE VRAIE PHOTO

Voici une vraie photo du lieu. Génère une version plus lumineuse et mieux cadrée, format 16:9, en gardant exactement les mêmes meubles, couleurs, matériaux et objets. N'ajoute aucun élément absent.

## 05 RETOUCHE LOCALE

Retouche uniquement [la zone]. [La correction]. Tout le reste de l'image reste strictement identique : lumière, perspective, objets.

# PETIT GLOSSAIRE

|                                 |                                                                                                              |
|---------------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Plan</b>                     | Une image fixe de ton voyage. Le film en enchaîne 6 à 14.                                                    |
| <b>Raccord</b>                  | Le passage d'un plan au suivant : zoom puis fondu.                                                           |
| <b>Échelle</b>                  | La taille du plan suivant dans le plan actuel. 0,70 = il occupe 70 % de la largeur.                          |
| <b>Plan intermédiaire</b>       | Un plan ajouté entre deux plans trop éloignés pour adoucir le zoom.                                          |
| <b>Retouche locale</b>          | Modifier une seule zone d'une image sans toucher au reste.                                                   |
| <b>Agrandissement (upscale)</b> | Augmenter la définition d'une image avec une IA spécialisée.                                                 |
| <b>Image clé</b>                | Une image complète dans une vidéo. Plus elles sont fréquentes, plus le déplacement dans la vidéo est rapide. |
| <b>crf</b>                      | Le réglage de qualité de l'encodage. Plus il est bas, plus la vidéo est nette et lourde.                     |
| <b>Sticky</b>                   | Un élément de page qui reste collé à l'écran pendant qu'on fait défiler.                                     |
| <b>Image d'attente (poster)</b> | L'image affichée avant que la vidéo soit chargée.                                                            |
| <b>Netlify</b>                  | Un hébergeur qui met un site en ligne gratuitement par glisser-déposer.                                      |

# FILM\_ZOOM.PY

```
#!/usr/bin/env python3
"""film_zoom.py · transforme une série d'images en film à zoom continu (effet « poupée russe »).

Chaque image doit être un détail de la précédente : la caméra plonge dans l'image 1,
arrive sur l'image 2, plonge dedans, et ainsi de suite.

Installation (une seule fois) :
    pip install opencv-python numpy
    et ffmpeg (https://ffmpeg.org, ou « brew install ffmpeg » sur Mac)

Utilisation :
    python3 film_zoom.py dossier_images film.mp4
    python3 film_zoom.py dossier_images film.mp4 --images-par-raccord 90 --largeur 1920

Les images sont prises dans l'ordre alphabétique : nomme-les 01.png, 02.png, 03.png...
"""
import argparse
import subprocess
import sys
from pathlib import Path

import cv2
import numpy as np

def charger(dossier, largeur, hauteur):
    fichiers = sorted(p for p in Path(dossier).iterdir() if p.suffix.lower() in {".png", ".jpg", ".jpeg", ".webp"})
    if len(fichiers) < 2:
        sys.exit("Il faut au moins deux images dans le dossier.")
    images = []
    for p in fichiers:
        im = cv2.imread(str(p))
        h, w = im.shape[:2]
        # recadrage au format de la vidéo, centré
        cible = largeur / hauteur
        if w / h > cible:
            nw = int(h * cible); im = im[:, (w - nw) // 2:(w - nw) // 2 + nw]
        else:
            nh = int(w / cible); im = im[(h - nh) // 2:(h - nh) // 2 + nh]
        images.append(cv2.resize(im, (largeur, hauteur), interpolation=cv2.INTER_AREA))
    return fichiers, images

def trouver_cadrage(a, b, default):
    """Où se trouve l'image b à l'intérieur de l'image a ? Renvoie (échelle, centre x, centre y) en proportions."""
    orb = cv2.ORB_create(4000)
    ka, da = orb.detectAndCompute(cv2.cvtColor(a, cv2.COLOR_BGR2GRAY), None)
    kb, db = orb.detectAndCompute(cv2.cvtColor(b, cv2.COLOR_BGR2GRAY), None)
    if da is not None and db is not None and len(ka) > 20 and len(kb) > 20:
        paires = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True).match(db, da)
        if len(paires) >= 30:
            pb = np.float32([kb[m.queryIdx].pt for m in paires])
            pa = np.float32([ka[m.trainIdx].pt for m in paires])
            M, ok = cv2.estimateAffinePartial2D(pb, pa, method=cv2.RANSAC, ransacReprojThreshold=4)
            if M is not None and ok.sum() >= 40:
                s = float(np.hypot(M[0, 0], M[1, 0]))
                h, w = a.shape[:2]
                cx, cy = M @ np.array([w / 2, h / 2, 1.0])
                if 0.25 < s < 0.95:
                    return s, cx / w, cy / h, "calé automatiquement"
    return default, 0.5, 0.5, "zoom central par défaut"

def lisse(x):
    x = min(max(x, 0.0), 1.0)
    return x * x * (3 - 2 * x)
```

# FILM\_ZOOM.PY (SUITE)

```
def raccord(a, b, s, cx, cy, n, sortie):
    h, w = a.shape[:2]
    bx, by = cx * w, cy * h
    # masque aux bords adoucis pour poser b dans a sans couture
    masque = np.zeros((h, w), np.float32)
    marge = int(0.08 * min(w, h))
    masque[marge:-marge, marge:-marge] = 1
    masque = cv2.GaussianBlur(masque, (0, 0), marge / 2)[..., None]
    for i in range(n):
        t = i / n
        z = s ** t                                # zoom exponentiel : vitesse perçue constante
        k = (1 - z) / (1 - s)                    # 0 au départ, 1 à l'arrivée
        vx, vy = w / 2 + (bx - w / 2) * k, h / 2 + (by - h / 2) * k
        # vue actuelle dans a
        Ma = np.float32([[1 / z, 0, w / 2 - vx / z], [0, 1 / z, h / 2 - vy / z]])
        vue = cv2.warpAffine(a, Ma, (w, h), flags=cv2.INTER_CUBIC, borderMode=cv2.BORDER_REFLECT)
        # b posé à sa place, visible de plus en plus nettement sur la fin du raccord
        alpha = lisse((t - 0.45) / 0.5)
        if alpha > 0:
            f = s / z
            Mb = np.float32([[f, 0, w / 2 + (bx - vx) / z - f * w / 2], [0, f, h / 2 + (by - vy) / z - f * h /
2]])
            pose = cv2.warpAffine(b, Mb, (w, h), flags=cv2.INTER_CUBIC)
            m = cv2.warpAffine(masque, Mb, (w, h))[..., None] * alpha
            vue = (vue * (1 - m) + pose * m).astype(np.uint8)
        sortie.write(vue.tobytes())

def main():
    ap = argparse.ArgumentParser(description="Film à zoom continu à partir d'une série d'images.")
    ap.add_argument("dossier"); ap.add_argument("film")
    ap.add_argument("--images-par-raccord", type=int, default=90, help="90 = 3 secondes à 30 images/s")
    ap.add_argument("--largeur", type=int, default=1920); ap.add_argument("--hauteur", type=int, default=1080)
    ap.add_argument("--zoom", type=float, default=0.78, help="zoom utilisé quand le calage automatique échoue")
    ap.add_argument("--qualite", type=int, default=18, help="18 = très haute qualité (master)")
    o = ap.parse_args()
    fichiers, images = charger(o.dossier, o.largeur, o.hauteur)
    ff = subprocess.Popen(["ffmpeg", "-loglevel", "error", "-y", "-f", "rawvideo", "-pix_fmt", "bgr24",
        "-s", f"{o.largeur}x{o.hauteur}", "-r", "30", "-i", "-",
        "-c:v", "libx264", "-preset", "slow", "-crf", str(o.qualite), "-pix_fmt", "yuv420p",
        "-movflags", "+faststart", o.film], stdin=subprocess.PIPE)
    for i in range(len(images) - 1):
        s, cx, cy, info = trouver_cadragage(images[i], images[i + 1], o.zoom)
        print(f"{fichiers[i].name} → {fichiers[i + 1].name} : échelle {s:.2f}, centre ({cx:.2f}, {cy:.2f}),
{info}")
        raccord(images[i], images[i + 1], s, cx, cy, o.images_par_raccord, ff.stdin)
    for _ in range(30):
        # une seconde de pause sur la dernière image
        ff.stdin.write(images[-1].tobytes())
    ff.stdin.close(); ff.wait()
    print("Film prêt :", o.film)

if __name__ == "__main__":
    main()
```

## PAGE\_SCROLL.HTML

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Mon site cinématique</title>
  <style>
    *{box-sizing:border-box;margin:0;padding:0}
    body{background:#050505;color:#F2EFE8;font-family:system-ui,sans-serif}
    /* La hauteur de .film règle la longueur du voyage : 400vh = 4 écrans de scroll. */
    .film{position:relative;height:400vh}
    .scene{position:sticky;top:0;height:100vh;height:100svh;overflow:hidden}
    .scene video,.scene img{position:absolute;inset:0;width:100%;height:100%;object-fit:cover}

    .titre{position:absolute;left:24px;right:24px;bottom:48px;opacity:0;transform:translateY(20px);transition:.6s}
    .titre.visible{opacity:1;transform:none}
    .titre h1,.titre h2{font-size:clamp(40px,8vw,96px);line-height:1;text-transform:uppercase}
    .suite{padding:80px 24px;max-width:720px;margin:0 auto;font-size:18px;line-height:1.6}
  </style>
</head>
<body>
  <div class="film" id="film">
    <div class="scene">
      
      <video id="video" muted playsinline preload="auto"></video>
      <!-- Chaque titre apparaît entre deux moments du voyage (0 = début, 1 = fin). -->
      <div class="titre" data-de="0" data-a=".25"><h1>Bienvenue</h1></div>
      <div class="titre" data-de=".3" data-a=".6"><h2>Entrez</h2></div>
      <div class="titre" data-de=".65" data-a="1.01"><h2>Installez-vous</h2></div>
    </div>
  </div>
  <section class="suite"><p>La suite de ton site : carte, horaires, contact...</p></section>

  <script>
    (function(){
      var
      film=document.getElementById('film'),video=document.getElementById('video'),affiche=document.getElementById('affiche');
      var titres=[];
      // Téléphone en portrait : on charge la version verticale.
      var mobile=innerWidth/innerHeight<.8;
      if(mobile)affiche.src='film-poster-mobile.jpg';
      video.poster=affiche.src;video.src=mobile?'film-mobile.mp4':'film-web.mp4';
      var cible=0,actuel=0,pret=false;
      function avancement(){var h=film.offsetHeight-innerHeight;return Math.min(1,Math.max(0,-
      film.getBoundingClientRect().top/h))}
      function maj(){
        var p=avancement();
        titres.forEach(function(t){t.classList.toggle('visible',p>=t.dataset.de&&p<+t.dataset.a)});
        if(pret)cible=p*(video.duration-.05);
      }
      // La vidéo rattrape le scroll en douceur : c'est ce qui donne l'effet « caméra ».
      function boucle(){
        if(pret&&Math.abs(cible-actuel)>.008){actuel+=(cible-actuel)*.2;video.currentTime=actuel}
        requestAnimationFrame(boucle);
      }
      video.addEventListener('loadedmetadata',function(){pret=true;maj()});
      // iPhone : une lecture très brève au premier toucher débloque le positionnement dans la vidéo.
      addEventListener('touchstart',function deb(){video.play().then(function(){video.pause()}).catch(function(){});removeEventListener('touchstart',deb)},{passive:true});
      addEventListener('scroll',maj,{passive:true});addEventListener('resize',maj);
      requestAnimationFrame(boucle);maj();
    })();
  </script>
</body>
</html>

```

À TOI DE TOURNER

# TON PREMIER FILM COMMENCE ICI

Commence petit : cinq plans, un lieu que tu connais, un week-end. Ton deuxième site sera deux fois plus rapide, ton troisième sera celui que tu montreras partout.

Une question, un blocage, un site à me montrer ? Écris-moi :  
[cadi.contactpro@gmail.com](mailto:cadi.contactpro@gmail.com) · [yaniscadi.fr/liens](http://yaniscadi.fr/liens)