Rapport de recherche

Maîtriser la multitude
Analyse comparative du workflow USD et de l'Alembic en simulation de foule

FROUIN Carl
ESMA Toulouse
2025-2026
Sous la supervision de Xavier ALIOT

TABLE DES MATIÈRES

I : « INTRODUCTION »....3
1.1- L'industrie face à l'explosion des données....3
1.2 - Alembic, un standard qui atteint ses limites....3
1.3 - Houdini et l'USD, un nouveau terrain d'expérimentation....3

II : « CONTEXTE SCOLAIRE ET QUESTIONNEMENTS »....5
2.1 - Le défi de la multitude....5
2.2 - Alembic vs USD, une rupture technologique....5
2.3 - La technique au service de la liberté artistique....5

III : « ÉTAT DE L'ART »....6
3.1 - Les limites des solutions traditionnelles de crowd....6
3.2 - Houdini : L'approche procédurale....6
3.3 - L'innovation USD Solaris, vers un pipeline non-destructif....7

IV : « HYPOTHÈSES ET OBJECTIFS »....8
4.1 - Hypothèse technique : réduire le volume de données....8
4.2 - Hypothèse artistique : préserver la liberté d'itération....8

V : « MÉTHODOLOGIE ET PROTOCOLE »....9
5.1 - L'environnement de production, un mini studio scolaire....9
5.2 - La farm Tractor, 272 PC et ses contraintes....9
5.3 - Un protocole adapté aux contraintes de production....9
5.4 - Du cas simple au cas réel, une approche par étapes....10

VI : « DESCRIPTION DE LA RECHERCHE »....11
6.1 - Partir d'un cube : valider chaque brique avant de complexifier....11
6.2 - Extension à la crowd, cinq variantes et quatre niveaux de détail....11
6.3 - Le cache d'animation, quand l'USD révèle toute sa puissance....12
6.4 - Variants, LOD et caméra culling, le contrôle total en LOP....13

VII : « RÉSULTATS »..... 14
7.1 - Export personnage isolé, un facteur de réduction de 122..... 14
7.2 - Temps d'export, de 2 minutes 43 à 5 secondes..... 15
7.3 - Géométrie des cinq variantes, 157 Mo contre 50 Mo..... 15
7.4 - Cache crowd 400 agents, 138 Mo contre 900 Go..... 16

VIII : « DISCUSSION, LIMITES ET CONCLUSION »..... 17
8.1 - Du cube aux 400 robots, l'écart entre théorie et réalité..... 17
8.2 - Les limites concrètes du workflow USD en production..... 17
8.3 - Un standard d'avenir, et une orientation confirmée..... 18

IX : « BIBLIOGRAPHIE »..... 19

REMERCIEMENTS

Je tiens en premier lieu à remercier mon superviseur, Xavier Aliot, pour son accompagnement tout au long de ce projet. Sa capacité à allier légèreté et exigence, vision décontractée et précision, en a fait un guide dont les conseils ont su parler à chacun. Son orientation m'a permis d'avancer avec confiance et le recul nécessaire.

Mes remerciements vont également à Arthur Justeau, camarade de la première heure, avec qui je partage cette aventure depuis le début de nos études. C'est ensemble que nous avons appris, progressé, et relevé les défis techniques les plus exigeants de ce film. Notre entraide mutuelle a été déterminante, en particulier sur les problématiques d'animation USD qui ont constitué le cœur de cette recherche. Avancer main dans la main sur un projet aussi ambitieux est une chance que je mesure pleinement.

I : « INTRODUCTION »

1.1- L'industrie face à l'explosion des données

L'industrie des effets visuels et de l'animation traverse aujourd'hui une période de mutation technologique sans précédent. Alors que les spectateurs exigent des mondes toujours plus vastes, plus denses et plus détaillés, les studios se heurtent à un défi de taille, la gestion de l'explosion des données numériques. Au cœur de cette problématique se trouve la simulation de foule (Crowd), l'un des domaines les plus exigeants techniquement. Qu'il s'agisse de peupler une ville, un stade ou une foule de concert de robots, le besoin de simuler des centaines, voire des milliers d'agents autonomes, pousse les pipelines dans leurs retranchements.

1.2 - Alembic, un standard qui atteint ses limites

Pendant plus d'une décennie, le format Alembic (.abc) a régné en maître pour l'échange de données géométriques. Cependant, sa nature "linéaire" et destructive où chaque modification impose souvent un ré-export complet de la scène devient un goulot d'étranglement majeur lors des phases de production intensive. Face à cette rigidité, l'émergence de l'**USD (Universal Scene Description)**, initialement conçu par les studios Pixar, propose un changement considérable. Plus qu'un simple format de fichier, l'USD se présente comme une structure de données non-destructive, permettant une collaboration simultanée entre départements et une gestion granulaire des itérations artistiques.

1.3 - Houdini et l'USD, un nouveau terrain d'expérimentation

Dans ce contexte, le logiciel Houdini de SideFX s'est imposé comme le terrain d'expérimentation idéal. Grâce à son architecture procédurale et son intégration native de l'USD via l'environnement Solaris (LOPs), il offre des outils de pointe pour la création de foules complexes. L'enjeu n'est plus seulement de réussir à générer un grand nombre d’agents dynamique, mais de pouvoir les modifier, les éclairer et les intégrer de manière fluide sans que la technique ne vienne entraver la vision artistique initiale.

Ce rapport de recherche se propose d'explorer les limites des méthodes conventionnelles face aux nouvelles possibilités offertes par l'USD. Nous chercherons à comprendre comment ce workflow transforme la gestion de données massives et s'il permet réellement d'offrir une plus grande souplesse d'itération aux artistes. À travers une série d'expérimentations techniques, allant du simple cube à la simulation de robots dynamiques, nous analyserons si l'USD constitue la réponse définitive aux défis de production actuels.

II : « CONTEXTE SCOLAIRE ET QUESTIONNEMENTS »

2.1 - Le défi de la multitude

Dans les productions actuelles, la simulation de foule est devenue une norme. Passer d'agents d'arrière-plan à des centaines de personnages "Hero" transforme radicalement les pipelines. Mon projet de fin d'études vise à animer plus de 500 robots dynamiques dotés d'une autonomie visuelle et physique crédible.

2.2 - Alembic vs USD, une rupture technologique

L'Alembic fonctionne comme une géométrie "cuite" : l'animation est figée. Entre Maya et Houdini, cette rigidité impose des ré-exports massifs à la moindre retouche, alourdissant le pipeline. À l'inverse, l'USD utilise une structure en couches (layers). En séparant l'animation du rigging, il permet de modifier les actifs sans ré-exporter les simulations, rendant le pipeline non-destructif et instantané.

2.3 - La technique au service de la liberté artistique

Le questionnement central reste la liberté créative : la technique est-elle un frein ? Un flux artistique efficace exige qu'une modification demandée puisse être visualisée rapidement. Si le pipeline impose des heures d'exportation, l'itération meurt. L'enjeu est de vérifier si l'écosystème USD sur Houdini permet de maintenir ce "flux" malgré la complexité technique, afin de concentrer l'énergie sur la mise en scène, l'aspect artistique plutôt que sur la gestion de cache disque.

C'est dans cette optique que ce rapport s'articule autour de la problématique suivante : “Le workflow USD en crowd sur Houdini surpasse-t-il les méthodes traditionnelles en termes de souplesse d'itération et de gestion de données ?”

III : « ÉTAT DE L’ART »

3.1 - Les limites des solutions traditionnelles de crowd

Le standard industriel (Golaem, Miarmy sur Maya) repose sur le "Baking" : la simulation est convertie en caches géométriques statiques (Alembic). Ce processus fige la donnée et brise le cycle d'itération.

- Explosion des données : L'étude de Bhalerao et al. (2024) démontre que le stockage linéaire (.abc) génère une croissance exponentielle du poids disque. L'Alembic "cuit" chaque sommet ou transformation par image, créant une redondance massive et saturant le stockage.
- Goulot d'étranglement : Selon Lallier (Pixar, 2018), les temps de lecture/écriture deviennent le frein principal en production. Toute retouche artistique (vitesse d'un agent, trajectoire) impose un recalcul total du cache, ce qui tue la réactivité nécessaire aux retours de supervision.
- Effet "Boîte Noire" : La documentation NVIDIA Omniverse souligne que les formats propriétaires limitent l'interopérabilité. Une fois exportée, la foule perd sa granularité, les agents intelligents deviennent de simples maillages passifs, rendant les ajustements de dernière minute techniquement impossibles sans un retour arrière coûteux vers le logiciel source.

3.2 - Houdini : L'approche procédurale

SideFX Houdini propose une alternative via son architecture d'Agents, où la foule n'est plus une géométrie, mais un flux de données.

- Optimisation mémoire : Contrairement à l'export polygonal, Houdini utilise des instances qui pointent vers des clips d'animation partagés. La documentation SideFX (2024) confirme que cette méthode minimise l'empreinte mémoire vive (RAM), permettant de manipuler des foules denses en temps réel dans le viewport.
- Contrôle par attributs : L'usage des VOPs/Wrangle permet de piloter la simulation via des scripts ou des nodes (triggers de comportement). Cette flexibilité est cruciale : elle permet d'isoler les données d'animation des données de collision ou d'apparence, préparant ainsi le terrain pour une structure de scène modulaire.

3.3 - L'innovation USD Solaris, vers un pipeline non-destructif

L'intégration de l'USD dans Solaris (LOPs) substitue les fichiers statiques par une structure de scène dynamique et hiérarchisée.

- Instancing et Variantes (NVIDIA, 2023) : L'USD ne charge qu'une seule géométrie source ; seules les matrices de transformation varient. L'utilisation des Variants permet en plus de changer le design d'un robot (couleurs, accessoires) sans charger de nouveaux fichiers, optimisant drastiquement le rendu.
- Layering et Composition Arcs : L'atout majeur est la séparation des départements. L'animation (SkelAnimation) réside sur une couche, tandis que le Lookdev réside sur une autre. Modifier une texture ne nécessite aucun ré-export : un simple override (couche d'écrasement) sur le "stage" suffit à mettre à jour. Toutes les scènes référençant le personnage bénéficieront de la modification instantanément.
- Validation via usdview : Cet outil de Pixar permet d'inspecter la hiérarchie et les "opinions" des couches en temps réel. Il garantit la propreté des données avant le rendu, évitant les erreurs de hiérarchie fréquentes lors des transferts entre départements.

Ressources et références clés

- Recherche Académique : Bhalerao et al. (2024) - Scalable Pipelines for Complex Scene Assemblage :

https://dl.acm.org/doi/full/10.1145/3721241.3733995

- Fondamentaux USD : Pixar Graphics Technologies - USD Introduction & Composition :

https://openusd.org/release/index.html

- Optimisation NVIDIA : NVIDIA Developer - OpenUSD Workflows and Performance :

https://developer.nvidia.com/openusd

- Documentation Technique : SideFX Houdini - Crowd Agents & Solaris LOPs

https://www.sidefx.com/docs/houdini/crowds/index.html

IV : « HYPOTHÈSES ET OBJECTIFS »

À partir de l'état de l'art et du contexte de production, deux hypothèses principales structurent cette recherche.

4.1 - Hypothèse technique : réduire le volume de données

La première hypothèse, d'ordre technique, postule que le workflow USD génère un volume de données significativement inférieur à celui de l'Alembic pour une animation équivalente, grâce à la séparation des couches de géométrie, de rig et d'animation.

4.2 - Hypothèse artistique : préserver la liberté d'itération

La seconde hypothèse, d'ordre artistique et organisationnel, avance que cette architecture non-destructive réduit les temps d'itération et préserve la liberté créative des équipes, en permettant des modifications sans ré-export complet des caches.

L'objectif est de valider ou d'invalider ces deux hypothèses à travers des expérimentations reproductibles, menées dans le cadre réel de la production du court métrage *Mortal Koil*, sur une foule de plus de 500 robots animés.

V : « MÉTHODOLOGIE ET PROTOCOLE »

Cette recherche s'appuie sur une méthodologie expérimentale à visée comparative, ancrée dans un contexte de production réelle. Elle combine des mesures techniques quantitatives (temps d'export, poids de fichiers, granularité des données) et une évaluation qualitative du pipeline (fluidité d'itération, lisibilité de la hiérarchie, indépendance des départements).

5.1 - L'environnement de production, un mini studio scolaire

Les expérimentations ont été menées dans un contexte scolaire qui s'apparente à un mini studio de production. L'école dispose d'une farm de rendu de 272 PC pilotée par Tractor 2.4, dont entre 190 et 200 machines sont actives et disponibles en moyenne, les autres étant éteintes ou indisponibles pour des raisons de licences ou de maintenance. Chaque machine dispose au minimum de 16 Go de RAM DDR4 et d'un processeur 12 cœurs, les configurations les plus puissantes montant jusqu'à 32 Go de RAM et 32 cœurs. Les licences RenderMan sont allouées dynamiquement en fonction du nombre de PC disponibles sur la farm, ce qui garantit une utilisation optimale des ressources sans goulot d'étranglement sur ce point.

5.2 - La farm Tractor, 272 PC et ses contraintes

Tractor 2.4 est une version ancienne du logiciel de gestion de farm, ce qui a pu occasionner quelques limitations dans les fonctions de communication et de contrôle des machines. Cela n'a pas impacté directement les résultats de la recherche mais représente une contrainte de confort non négligeable en production quotidienne.

5.3 - Un protocole adapté aux contraintes de production

En termes de performances, un PC lambda de la farm met en moyenne 10 minutes à ingérer la crowd de 400 agents en mémoire vive, puis environ 15 minutes à rendre une image complète. Le principal goulot d'étranglement ne réside donc pas dans le calcul du rendu lui-même, mais dans la phase de chargement des données USD

en RAM une réalité directement liée aux configurations minimales des machines disponibles.

5.4 - Du cas simple au cas réel, une approche par étapes

Le protocole initial prévoyait une comparaison progressive et scalable entre les deux approches : exporter 10, 50 puis 400 agents animés depuis Maya via Alembic, puis reproduire le même exercice dans Houdini via USD, afin d'établir une courbe de comparaison claire à chaque palier de complexité.

Cependant, les contraintes réelles de la production, planning serré, deadlines de rendu et besoins immédiats du film *Mortal Koil*, ont imposé une réorientation du protocole. Cette adaptation est en soi une donnée méthodologique importante, dans un contexte de production réelle, la démarche idéale cède souvent la place à une approche plus pratique, construite chemin faisant, par itérations successives plutôt que par protocole figé.

L'expérimentation s'est donc déroulée naturellement, en testant des exports USD organisés en couches distinctes (géométrie, rigging, animation), en assemblant des agents de crowd complets, et en poussant progressivement la complexité jusqu'aux conditions réelles de la production. Une comparaison directe avec des exports Alembic équivalents a ensuite été établie sur plusieurs cas de test, du personnage isolé à la simulation de foule multi-agents, sur des plages temporelles représentatives du film.

Les indicateurs retenus sont d'ordre technique (temps d'export, poids des fichiers, granularité par couche) et qualitatif (capacité d'itération sans ré-export global, stabilité du pipeline, lisibilité de la hiérarchie pour l'ensemble de l'équipe).

VI : « DESCRIPTION DE LA RECHERCHE »

6.1 - Partir d'un cube : valider chaque brique avant de complexifier

L'expérimentation a débuté par une approche volontairement minimaliste. Avant de toucher au moindre personnage du film, les premiers tests ont été conduits sur des géométries basiques : un cube, puis deux cubes, puis une dizaine d'objets simples. Cette logique de dégrossissement progressive est fondamentale, elle permet de valider chaque brique du système de façon isolée, sans que la complexité de l'asset ne vienne masquer une éventuelle erreur de structure. Ce principe, revenir au plus simple dès que quelque chose ne fonctionne pas, a guidé l'ensemble de la démarche et s'est révélé systématiquement plus efficace que de tenter de débugger une scène complexe.

Une fois le comportement des couches USD compris et validé sur ces cas simples, la complexité a été augmentée progressivement : la main du robot, puis son buste, puis le personnage complet avec ses 2 millions de polygones. À chaque étape, le même protocole a été appliqué : exporter les trois couches séparément, les assembler en LOPs, valider visuellement, puis passer à l'étape suivante uniquement si le résultat était propre.

C'est au cours de cette phase qu'une observation centrale a émergé : le layer d'animation, qui ne contient aucune géométrie, ne stocke que les données de transformation des joints. Son poids est donc découplé de la complexité du mesh. Cette propriété change radicalement la logique d'itération : corriger une animation ou mettre à jour un cycle de marche ne nécessite de ré-exporter que ce seul fichier, pendant que la géométrie et le rig restent intacts et partagés entre tous les shots du film.

6.2 - Extension à la crowd, cinq variantes et quatre niveaux de détail

Fort de cette validation sur un personnage isolé, le protocole a été étendu aux conditions réelles de la simulation de foule. La crowd de *Mortal Koil* repose sur cinq variantes de robots (crowdVar01 à crowdVar05), chacune disposant de quatre niveaux de détail (LOD0 à LOD3). Cette architecture a été entièrement exportée en USD depuis Maya, produisant pour chaque variante deux fichiers distincts : un layer de modélisation et un layer de rigging.

Les poids obtenus sur les layers de modélisation USD sont les suivants : crowdVar01 pèse 13,4 Mo, crowdVar02 8,9 Mo, crowdVar03 10,5 Mo, crowdVar04 8,1 Mo et crowdVar05 9,5 Mo, soit un total d'environ 50,5 Mo pour l'ensemble de la géométrie des cinq variantes. Cette légèreté s'explique par une décision d'architecture importante : lors de l'export USD, seule la géométrie brute est écrite. Aucun attribut custom, aucune primvar additionnelle, aucune donnée de simulation ne vient alourdir ces fichiers. Toutes ces informations sont gérées séparément, dans d'autres couches dédiées, ce qui maintient les fichiers de géométrie aussi légers et réutilisables que possible.

Les layers de rigging, qui ne contiennent que les relations joints/géométrie, sont particulièrement légers : de 278 Ko pour crowdVar02 à 893 Ko pour crowdVar01, pour un total inférieur à 2,3 Mo. À titre de comparaison, les scènes Maya source de ces mêmes personnages dépassent en moyenne 300 Mo par fichier .ma, sans même inclure de cache d'animation exportable.

La même comparaison menée en Alembic révèle des écarts significatifs sur la géométrie seule, avant même d'inclure l'animation : crowdVar01 atteint 33,8 Mo, crowdVar02 30,5 Mo, crowdVar03 36 Mo, crowdVar04 26,2 Mo et crowdVar05 31,3 Mo, soit un total de 157,8 Mo pour la géométrie statique uniquement, contre 50,5 Mo en USD.

6.3 - Le cache d'animation, quand l'USD révèle toute sa puissance

L'étape suivante a consisté à exporter le cache d'animation de la crowd complète depuis Houdini SOPs. Ce cache, généré pour 400 agents animés sur 70 frames, ne contient aucune géométrie : il stocke exclusivement les données de SkelAnimation de chaque agent, c'est-à-dire les transformations de joints frame par frame, accompagnées d'un ensemble de métadonnées incluant l'identifiant de l'agent, son état de naissance, et plus d'une quarantaine d'attributs primvars personnalisés. Ces attributs, assignés par agent, encodent des informations exploitables en aval par les départements de shading et de lighting : variation de peinture, couleur, indice de variante, état d'usure, et d'autres paramètres procéduraux propres à la direction artistique du film.

Le poids total de ce cache pour 400 agents sur 70 frames est de 138,6 Mo. Par comparaison, une extrapolation d'un export Alembic classique sur le même volume d'agents, en appliquant le ratio observé sur le personnage isolé, aboutit à un cache estimé à plusieurs centaines de gigaoctets, rendant ce workflow inenvisageable dans un contexte de production avec un stockage réseau partagé et limité.

6.4 - Variants, LOD et caméra culling, le contrôle total en LOP

Une fois le cache importé dans Houdini Solaris, plusieurs fonctionnalités propres à l'USD ont été exploitées pour affiner la gestion de la crowd en LOP.

Le système de Variant Sets a permis d'associer à chaque agent l'une des cinq variantes géométriques sans dupliquer aucun fichier source : chaque agent pointe vers un prototype unique et hérite de la variante. Cette approche garantit que modifier le mesh d'une variante se répercute instantanément sur tous les agents qui l'utilisent, sans aucun ré-export.

Les niveaux de détail ont été gérés via un second Variant Set LOD, activé dynamiquement en fonction de la distance à la caméra. Les agents en arrière-plan chargent ainsi une version allégée de leur géométrie, réduisant significativement la charge mémoire au moment du rendu sans impacter la qualité visuelle au premier plan.

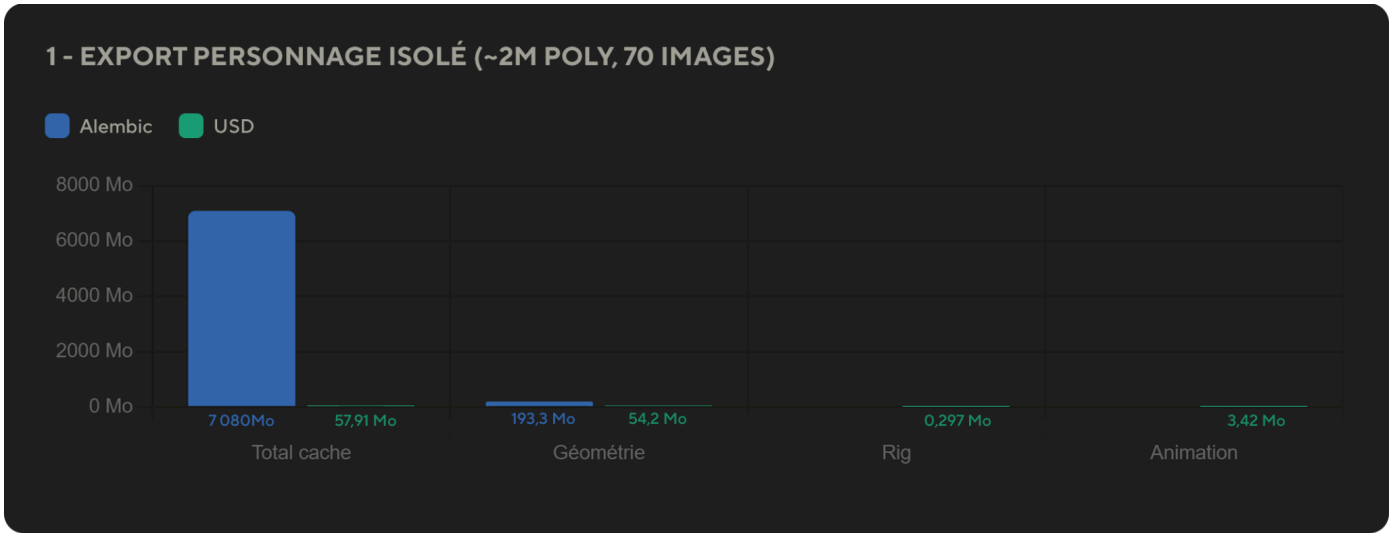
Enfin, un système de caméra frustum culling a été mis en place en Python LOP, à partir des paramètres de la caméra de shot, un script calcule les plans du frustum et désactive automatiquement les agents situés hors champ en leur appliquant un prune (désactivation) dans le stage. Cette opération, purement non-destructive, ne supprime aucune donnée et peut être annulée ou recalculée à tout moment sans modifier les fichiers source.

L'ensemble de ces mécanismes, variants, LOD et culling, opère exclusivement au niveau du stage USD en LOP, sans jamais nécessiter de ré-export des caches sous-jacents. C'est précisément cette capacité à empiler des décisions artistiques et techniques sous forme de couches légères et réversibles qui distingue fondamentalement ce workflow des approches traditionnelles basées sur des caches géométriques figés.

VII : « RÉSULTATS »

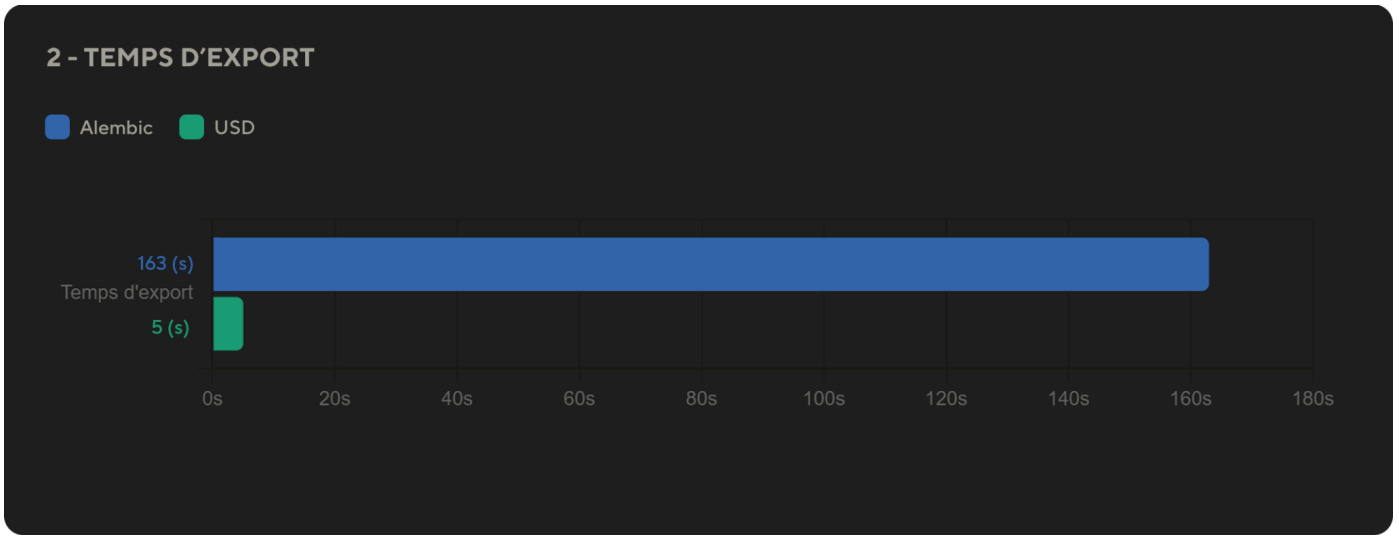
Les mesures recueillies confirment sans ambiguïté les deux hypothèses formulées en amont, l'USD surpasse l'Alembic aussi bien sur le volume de données que sur la souplesse d'itération.

7.1 - Export personnage isolé, un facteur de réduction de 122



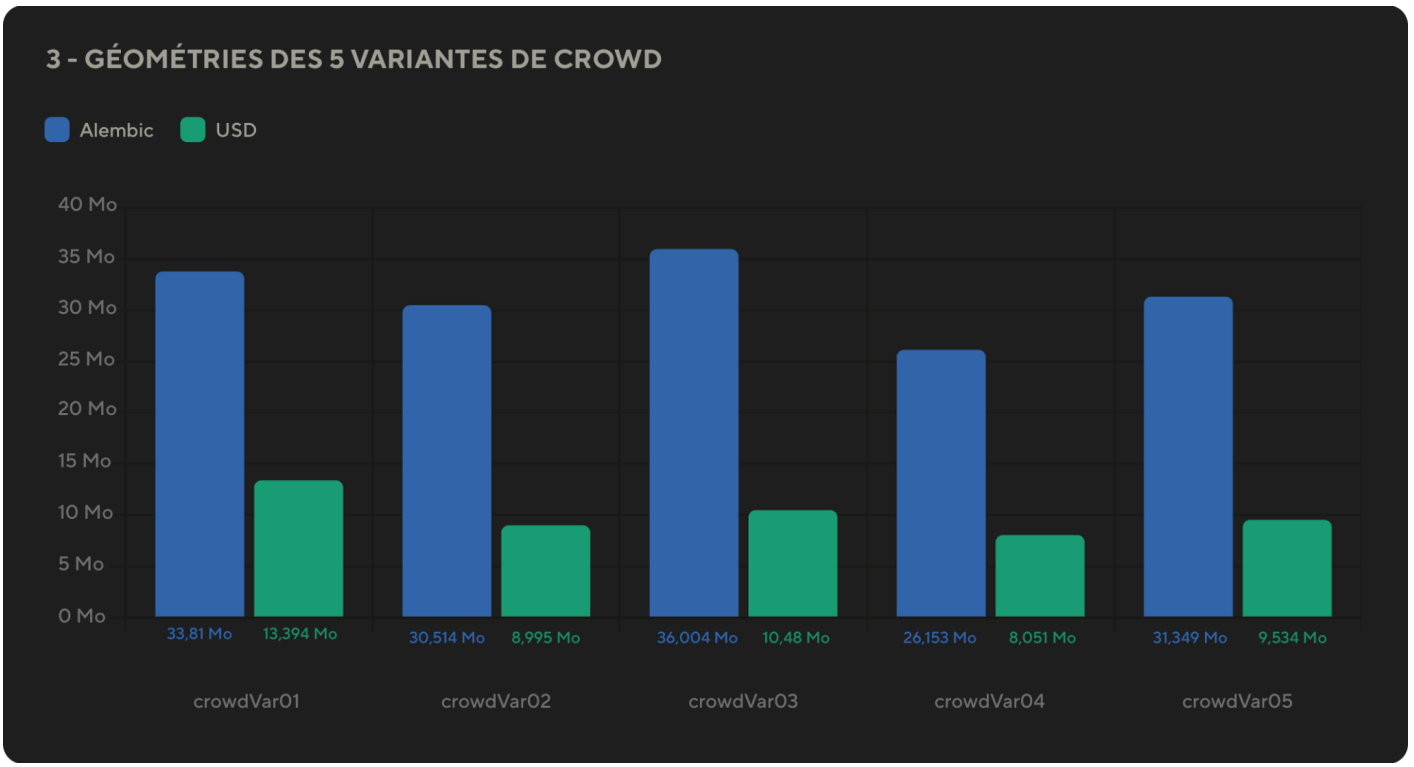
Sur un personnage isolé, le cache Alembic atteint 7,08 Go contre 57,91 Mo en USD, soit une réduction d'un facteur 122. L'Alembic écrit l'intégralité des positions de vertices à chaque frame, tandis que l'USD ne stocke que les transformations de joints dans un layer dédié de 0,297 Mo, la géométrie et le rig étant partagés entre tous les shots.

7.2 - Temps d'export, de 2 minutes 43 à 5 secondes



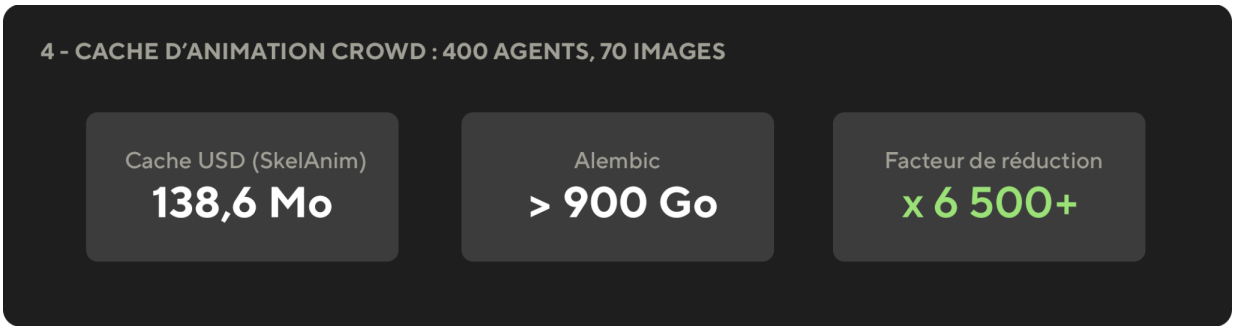
5 secondes en USD contre 2 minutes 43 en Alembic sur le même cas de test. En production intensive, cette différence transforme directement la capacité d'itération des équipes : une correction d'animation devient quasi instantanée là où l'Alembic impose une attente systématique.

7.3 - Géométrie des cinq variantes, 157 Mo contre 50 Mo



Sur la géométrie statique des cinq variantes, l'écart est net : 157,8 Mo en Alembic contre 50,5 Mo en USD. Ce résultat s'explique par le fait que l'export USD ne conserve que la géométrie brute, sans aucun attribut ou primvar superflue, là où l'Alembic embarque l'intégralité des données du mesh source.

7.4 - Cache crowd 400 agents, 138 Mo contre 900 Go



Le résultat le plus significatif concerne le cache de simulation complet. Pour 400 agents animés sur 70 frames, le cache USD pèse 138,6 Mo. L'équivalent Alembic dépasse 900 Go, un volume totalement incompatible avec un stockage réseau partagé en contexte étudiant, et qui rendrait ce workflow inenvisageable en production.

VIII : « DISCUSSION, LIMITES ET CONCLUSION »

Les résultats obtenus valident les hypothèses de départ. Le workflow USD surpasse les méthodes traditionnelles sur les deux axes étudiés, en volume de données et en souplesse d'itération. Ces résultats doivent cependant être nuancés par l'expérience concrète de leur mise en œuvre.

8.1 - Du cube aux 400 robots, l'écart entre théorie et réalité

Sur le papier, la logique USD est élégante. Appliquée à un cube, elle fonctionne immédiatement. Appliquée à 400 agents dotés de rigs complexes issus de Maya, elle révèle une réalité bien plus exigeante : skins mal liés, matrices de transformation incorrectes, écarts d'échelle entre logiciels, erreurs de composition de layers. La méthode de dégrossissement progressive s'est révélée indispensable : ne complexifier qu'une fois chaque brique validée, et simplifier dès qu'un problème résiste.

L'USD est un langage profond. Pour des usages basiques, il est immédiatement efficace. Pour des usages avancés, il demande une compréhension fine de sa logique interne que peu de ressources documentaires permettent d'acquérir rapidement. C'est à la fois sa plus grande force et ce qui le rend parfois vertigineux.

8.2 - Les limites concrètes du workflow USD en production

La principale limite du workflow USD s'est fait ressentir lors de l'assemblage de plans complexes avec de nombreuses références imbriquées. Contrairement à l'Alembic qui écrase toutes les informations dans un fichier unique et donc rapide à lire, l'USD doit résoudre l'ensemble des couches référencées au moment du chargement, ce qui peut rendre le viewport très long à charger sur des scènes denses. Cette limite est directement liée à la vitesse du serveur réseau local et aux configurations minimales des machines, notamment les 16 Go de RAM DDR4 des PC les moins puissants, qui peinent à ingérer une scène crowd complète en temps interactif. Des solutions ont été mises en place pour contourner ce problème, comme la création de layers d'assemblage allégés pour le travail quotidien, ne chargeant que 50 agents au lieu de 400, les 400 étant envoyés en rendu complet sur la farm.

La version de Tractor utilisée a également représenté une limite de confort, avec des fonctions de communication et de contrôle des machines plus restreintes que les versions actuelles, sans toutefois impacter directement les résultats de la recherche.

8.3 - Un standard d'avenir, et une orientation confirmée

Cette recherche a confirmé que l'USD constitue une réponse concrète aux défis posés par la simulation de foule, non pas parce qu'il simplifie le problème, mais parce qu'il offre les outils pour le maîtriser proprement. C'est dans l'écart entre comprendre l'USD sur un cas simple et l'appliquer à une foule de 400 robots que réside l'essentiel de l'apprentissage. Ce projet a ouvert des perspectives vers des rôles de Pipeline TD ou FX TD, des métiers où rigueur technique et capacité à transmettre sont aussi importantes que le savoir-faire lui-même.

IX : « BIBLIOGRAPHIE »

ACM Digital Library SIGGRAPH Course ‘25 - USD in Production
https://dl.acm.org/doi/full/10.1145/3721241.3733995

NVIDIA Developer (2025). *OpenUSD Workflows and Performance*. Consulté en 2026. https://developer.nvidia.com/openusd

NVIDIA Omniverse (2025). *Universal Scene Description — Interoperability and Pipeline Integration*. Consulté en 2026.
https://docs.omniverse.nvidia.com/usd/latest/index.html

Pixar Graphics Technologies (2025). *USD Documentation*. Consulté en 2026.
https://openusd.org/release/index.html

SideFX (2025). *Houdini Crowd Agents Documentation*. Consulté en 2026.
https://www.sidefx.com/docs/houdini/crowds/index.html

SideFX (2025). *Houdini Solaris & USD Basics*. Consulté en 2026.
https://www.sidefx.com/docs/houdini/solaris/index.html