

VegasMaskTracker : VEGAS ベジェマスキング → Moho 連携ガイド

Weaver Pipeline Series 公式完全活用マニュアル

目次

1. はじめに（開発背景と本ツールの目的）
2. 3大連携モードの徹底比較と用途マップ
3. 二大出力データ「CSV」と「画像シーケンス」の役割分担と可能性
4. VEGAS Pro 側での操作手順（完全ガイド）
5. Moho 側でのインポートと設定（完全ガイド）
6. 実践ワークフロー&チュートリアル（Step-by-Step）
 - チュートリアル 1：看板・壁面への 3D パースハメ込み（Smart Warp Mesh）
 - チュートリアル 2：キャラクターへの小物・フキダシ 2D 追従（Bone Rig）
 - チュートリアル 3：手ブレ実写動画へのアニメ的エフェクト作画（Stabilize Video）
 - チュートリアル 4：実写の手の後ろをアニメキャラが通るオクルージョン合成（画像シーケンス活用）
7. Moho 内部レイヤー構造と非破壊作画保護システム
8. 座標正規化と技術仕様（数学的背景と Fail-Safe）
9. トラブルシューティング&よくある質問（FAQ）

1. はじめに（開発背景と本ツールの目的）

課題と解決

- **Moho 標準トラッカーの限界:** Moho に内蔵されているポイントトラッカーは、被写体の回転や手ブレ、被写体同士の交差や一時的な遮蔽（オクルージョン）が発生すると追従ポイントが外れやすいという弱点があります。
- **VEGAS Pro / Mocha VEGAS の強力なトラッキング性能:** VEGAS Pro の「ベジェマスキング」（および内部の Mocha トラッカーエンジン）は、平面検出（Planar Tracking）技術により、被写体の移動・回転・拡大縮小、さらには 3D 空間での遠近法（パースペクティブ歪み）を高精度かつ極めて安定して追従できます。
- **従来の壁を突破した VegasMaskTracker:** 従来、VEGAS のスライダー経由でキーフレームを書き出すと「台形パース歪み（Corner Pin）」が切り捨てられてしまい、2D 平行移動に縮退してしまうという壁がありました。本ツールは、**C# ネイティブ幾何解析アルゴリズム（凸包+4頂点縮約+最近傍追従）および Mocha パラメータの直接高速抽出**によりこの壁を完全突破。外部の Python や中間変換ツールを一切介さず、**VEGAS Pro から直接一発で CSV を出力し、Moho へ 100% 自動同期**させるプロ仕様のパイプラインを確立しました。
- **ビデオ手ブレ補正（フリーズモーション / 画面固定）連携:** 手持ちカメラ等で激しく揺れる実写動画に対し、Moho 内で特定被写体を中心にピタッと静止させる「インバーススタビライズ」を実装。「揺れる動画の被写体を画面中央にフリーズさせ、静止画を描く感覚で Paint_Canvas 上に手描きイラストやアニメエフェクトを描き込む」という、全く新しいハイブリッド制作ワークフローを実現しました。

[!TIP] 【連携の基本思想：VEGAS側で意図を選び、Moho側で素直に受ける】 本システムの連携は極めてシンプルです：

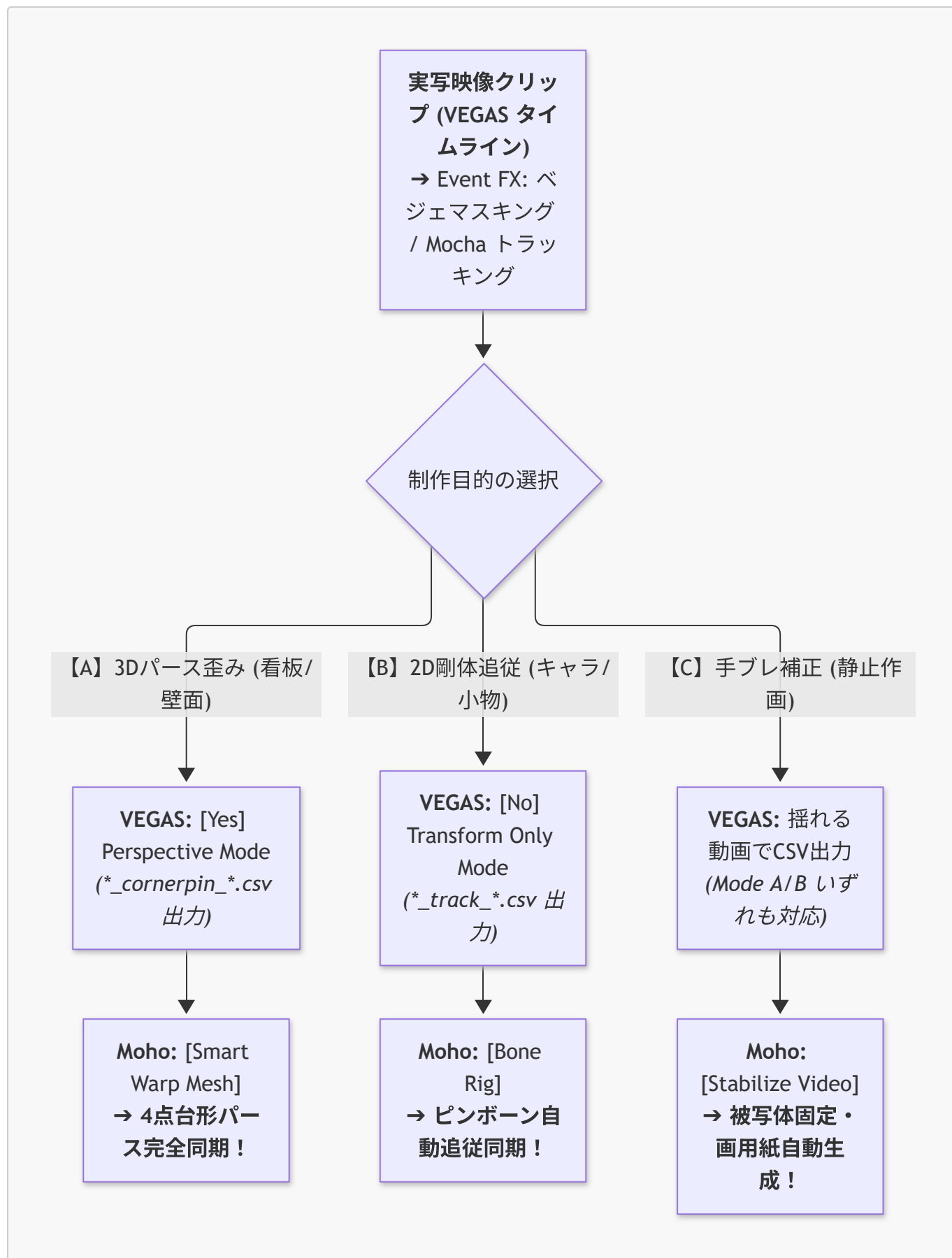
1. VEGAS Pro 側で「3D パース歪み」か「2D 追従」かを選んで書き出す。
2. Moho 側で、その CSV を「Smart Warp Mesh」「Bone Rig」「Stabilize Video」のいずれか目的の構造に流し込む。面倒な座標計算やキーフレームの手打ちは一切不要となり、実写とアニメーションの融合が瞬時に実現します。

2. 3大連携モードの徹底比較と用途マップ

VegasMaskTracker は、映像制作のあらゆるシチュエーションに応える **3つの連携モード** を備えています。

モード	VEGAS側の出力選択 (Exportスクリプト)	生成される CSV	Moho側の選択 (Trackerツール)	Moho上での仕上がり	主な用途
A. 3Dパース (4コーナー)	[Yes] Perspective Mode	*_cornerpin_*.csv (# Type: CornerPin)	[Smart Warp Mesh]	4頂点のSmart Warpメッシュが台形パースに歪みながら追従	看板、ポスター、床、壁面、スマホ画面
B. 2D追従 (ピンボーン)	[No] Transform Only Mode	*_track_*.csv (# Type: Transform)	[Bone Rig]	素材の形状を保ったまま、1本のPin Boneが位置・回転・均等拡縮を追従	キャラクター、名札、バッジ、吹き出し、小物
C. 手ブレ補正 (画面固定)	[Yes] または [No]	どちらの CSV でも対応可能	[Stabilize Video]	揺れる元動画に逆変換をあてて被写体を中央静止。 Paint_Canvas で作画	被写体への直接ペイント、アニメ的エフェクト

パイプライン連携フロー図



3. 二大出力データ「CSV」と「画像シーケンス」の役割分担と可能性

`Export_VegasMask_CSV.cs` は、「**CSVファイル**」だけでなく、オプションで「**画像シーケンス (マスクPNG連番)**」を同時に出力する機能を備えています。

[!IMPORTANT] 「4頂点自動化が完成した今、なぜ画像シーケンスを残しているのか？」その答えは、「4頂点メッシュ（四角形）では絶対に不可能な、複雑な有機的形狀の切り抜きと前後合成（オクルージョン）を可能にするため」です。

データ種別	出力形式	主な役割	得意な表現
CSV データ	テキスト (CSV)	板状・平面の変形貼り付け & ボーン追従	看板、ポスター、スマホ画面、名札、フキダシ
画像シーケンス	連番画像 (PNG)	被写体の有機的輪郭による切り抜き & 前後関係（オクルージョン）	実写の手や指の後ろにキャラを隠す、人物シルエット塗りつぶし、輪郭発光

Q. Mocha で作るマスクは「4点」でなくてもいいの？ 手の形でも4点が出る？

A. はい！ Mocha のマスクは 4 点である必要は一切ありません！ 何十点もある自由なスプラインで手の指の形をなぞって大丈夫です。

- **手の形にした場合の画像シーケンス（PNG）**：指の1本1本まで、Mocha でなぞった通りの複雑な「手のシルエットそのもの」が白マスク連番画像として出力されます。
- **手の形にした場合の CSV データ（4点）**：C# スクリプトに組み込まれた幾何学アルゴリズム（凸包＋多角形縮約）が働き、手の輪郭全体をすっぽり包み込むような**「外枠のバウンディング四角形（最も特徴的な4隅）」を自動計算して 4 点 CornerPin として出力**します。

画像シーケンスの 4 大実践テクニック

1. **実写被写体との前後関係合成（オクルージョン処理）**：「実写の手の後ろをアニメキャラが通り、指の形に沿ってキャラが隠れる」という表現が、Moho のレイヤーマスキングで一発で実現します（後述のチュートリアル 4 参照）。
2. **有機的シルエットのテクスチャ流し込み / 影人間表現**：人物の全身マスクを抽出し、実写人物の形そのままに黒塗りの影にしたり、Moho のアニメ調パターン柄で埋め尽くす演出が可能です。
3. **輪郭発光・電撃オーラジェネレーター**：二値（白黒）マスク画像を Moho や外部ツール（AE / Krita）のエッジ抽出・ブラーにかけることで、実写被写体の周囲にパチパチと弾けるオーラやネオン輪郭を自動生成できます。
4. **超軽量オニオンスキン（作画ガイド）**：重たい 4K 実写動画を非表示にし、白マスク連番だけをガイド（Moho 上でレイヤー不透明度を下げて半透明化可能）にして軽快にペン作画を進めることができます。

4. VEGAS Pro 側での操作手順（完全ガイド）

[!IMPORTANT] **対応バージョン**: VEGAS Pro 21 ~ 23 (Mocha VEGAS / ベジェマスキング搭載環境)。
※ VEGAS Pro 23 まで動作検証・公式サポート対象となります（将来バージョンでのAPI仕様変更等は保証対象外）。

ステップ 1：トラッキングの実行（Mocha VEGAS / ベジェマスキング）

1. VEGAS Pro のタイムライン上で、対象のビデオイベント（クリップ）の **Event FX** アイコンをクリックします。
2. **Mocha VEGAS**（または **ベジェマスキング**）を追加して起動します。

3. 追従させたい領域（看板、人物の顔、手、壁面など）をスプラインツールで囲み、トラッキング（Track Forward）を実行します。
4. トラッキング完了後、保存して Mocha ウィンドウを閉じます。

[!IMPORTANT] ★【超重要】Mocha VEGAS 使用時の必須設定：View Matte にチェックを入れる！

Mocha ウィンドウを閉じた後、VEGAS の Event FX ウィンドウ内にある「View Matte」に必ずチェック [X] を入れてください。これにより、プレビュー画面に**「黒背景に純白のマスク」**が表示される状態になります。本ツールの CornerPin 解析エンジンは、この純白マット領域を 1 ピクセル単位で読み取って真の 4 コーナーを数学的に極めて安定して抽出します。（※ View Matte がオフのままだと、画像が抽出できず「Mask region was not detected」エラーになります）

[!NOTE] ★ 3D 遠近法（台形パース）を最高精度で取得する極意: Mocha 画面上部の「Show Planar Surface（青い四角形 [S] アイコン）」をオンにし、四角形マスクの角を Planar Surface の 4 隅にぴったり合わせておくと、Mocha の計算した 3D 平面パースと 100% 一致するマスクが描画されます。

[!TIP] ★【実践テクニック】VEGAS Pro 22 標準のモーショントラック / ベジェマスキングから 4 点（CornerPin）を取り込む方法: VEGAS Pro 22 のタイムラインで「モーショントラッキング（遠近感/Perspective）」を実行、または標準の「ベジェマスキング」を使用する場合、実写映像のまま書き出すと被写体の影や明暗差がノイズとなり、マスクの角が欠落して座標が暴れる原因になります。以下の手順で「純白マット」を作成して書き出すことで、Mocha と同等の高精度な 4 点 CornerPin が抽出できます：

1. トラックの上（または別トラック）に、メディア ジェネレーター → 単色（Solid Color）→ 白（White）のイベントを動画と同じ長さで配置します。
2. その白イベントの Event FX に VEGAS ベジェマスキングを追加し、モーショントラックデータ（遠近感）をペーストまたは適用します（四角形マスク）。
3. これにより、プレビュー画面が Mocha の View Matte と全く同じ**「黒背景に純白の四角形マスク」**になります。
4. この白イベントを選択して Export_VegasMask_CSV.cs → [Yes] Perspective Mode を実行すると、実写のノイズが 100% 排除された完璧な 4 点 CornerPin CSV が出力されます。

ステップ 2：Export_VegasMask_CSV.cs での一発書き出し

1. タイムライン上で対象のクリップ（ビデオイベント）をクリックし、選択状態（黄色い枠）にします。
2. VEGAS のトップメニューから ツール（Tools）→ スクリプト（Scripting）→ スクリプトの実行（Run Script...）を選択します。
3. Export_VegasMask_CSV.cs を選択して実行します。
4. モード選択ダイアログ：
 - [Yes] = Perspective / CornerPin Mode（3Dパース・台形歪み・Smart Warp Mesh連動）：1ピクセル完全解像度走査 + 最大面積内接四角形（MIQ: Maximum Inscribed Quadrilateral）幾何アルゴリズムにより、Mocha の白マスクから真の 4 コーナーを高精度に自動特定し、CornerPin CSV を生成します。間引きによるブレや角のジッターは完全に排除され、Moho のメッシュ頂点がマスクに完璧に吸着します。
 - [No] = Transform Only Mode（2D 位置・回転・スケール）：スライダーパラメータから高速に Transform CSV を生成します（極小の被写体やボーン追従に最適）。
5. 画像シーケンス保存オプション（Perspective モード時のみ）：

- **[No] (★推奨・爆速)** : 連番 PNG をディスクに残さず、メモリ上だけで 4 隅を高速抽出して CSV のみを出力します (数秒で完了)。
- **[Yes]** : オクルージョン合成 (100% 透過の型抜き・減算マスク) や作画ガイドとして使いたい場合に、白マスク (Alpha 255) の PNG 連番フォルダを同時に保存します。

6. ファイル名と CSV の保存 (テイク名 & レイヤー名連動) :

- スクリプトがタイムライン上の **テイク名 (クリップ名)** を自動取得し、初期ファイル名として提案します (例: `vegas_mask_cornerpin_TakeName.csv` または `vegas_mask_track_TakeName.csv`)。
- Mocha 内のレイヤー名があれば末尾に付加されます (例: `_看板A`)。
- **解像度フリーの自動正規化**: 映像が 4K、Full HD、縦動画などいかなる解像度であっても、スクリプトがプロジェクト設定から自動計算し、Moho 空間用の正規化座標 (0.0~1.0) へ完璧に変換して保存します。

5. Moho 側でのインポートと設定 (完全ガイド)

ステップ 1: インポート前の下準備

- **モード A (Smart Warp Mesh) または モード B (Bone Rig) の場合**: レイヤーパレットの選択状態はどこにあっても構いません。深い階層に潜っていても、スクリプトが自動的にルート階層へ安全に新しいグループを配置します。
- **モード C (Stabilize Video : 手ブレ補正) の場合**: あらかじめ Moho タイムライン上に読み込んである **動画レイヤー (または連番画像レイヤー)** を選択した状態でスクリプトを実行してください。選択した動画が自動的に手ブレ補正グループ (`Video_Stabilizer`) の内側に格納されます。

ステップ 2: スクリプトの起動と CSV 選択

1. Moho Pro のツールパレットにある **「VegasMaskTracker」** ツールアイコンをクリックします (またはメニュー `スクリプト > Weaver > VegasMaskTracker: Import & Bake` を実行)。
2. ファイル選択ダイアログで、VEGAS から書き出した CSV を選択します。
 - **日本語パスの完全サポート**: Windows 環境でフォルダ名やファイル名に日本語が含まれていても、`SafeReadText` 機構により文字化けせず確実に読み込まれます。
 - **二重防御 (Dual-Defense) フェイルセーフ**: 万が一古いピクセル座標形式の CSV を読み込んだ場合でも、スクリプトが自動検知して正規化座標へ補正するため、オブジェクトが画面外へすっ飛ぶ事故を防止します。

ステップ 2.5: フレームレート不一致 (FPS Mismatch) の自動検出

Moho プロジェクトの FPS (例: 24 fps) と CSV の映像 FPS (例: 29.97 fps) が異なる場合、自動的に警告ダイアログが表示されます:

- **[Auto-Sync (Resample)] (★推奨)** : CSV の実時間 (`TimeSec`) に基づき、Moho タイムラインの対応フレームへ自動リマップしてキーフレームをベイクします。作画を 24 fps で行いながら、実写の 29.97 fps と 100% 完璧に同期します。
- **[Abort (Change FPS)]** : インポートを中断します。Moho の `ファイル > プロジェクト設定` で FPS を揃えてから再実行したい場合に使用します。
- **[Ignore (1:1 Frame)]** : 時間計算を行わず、CSV のフレーム番号をそのまま 1:1 で Moho タイムラインに打ち込みます。

ステップ 3：出力先構造と各モードの詳細設定

CSV を読み込むと、目的の構造を選択するダイアログが表示されます。

【A. Smart Warp Mesh モード設定】

(※ CornerPin CSV 読み込み時はもちろん、Transform CSV 読み込み時にも選択可能です)

- **Target Group Name (対象グループ名)** : テイク名やレイヤー名から自動提案されます (例: Vegas_Warp_看板A)。
 - ★ **既存名 (Update)** : グループ内に描いたイラストレイヤー (Paint_Canvas 等) は一切削除・破壊されず、キーフレームのみが最新データに上書き更新されます (完全な作画保護!)。
 - ★ **Frame 0/1 変形ジャンプの完全根絶**: Frame 1 の追跡四角形座標が Frame 0 のレストポーズおよびベースキーフレームに直接適用されるため、1フレーム目に進んだ瞬間にメッシュが急激にひし形などに歪む現象は 100% 完全に消滅しています。
- **Smart Warp Mesh Margin Multiplier (マージン倍率)** : マスク領域に対して、作画キャンバスをどのくらいの広さで作成するかをラジオボタンで選択します：
 - () **1.0x (マスク境界等倍 - 余白なし)**: 看板やポスター、スマホ画面など、四角形の枠内にぴったり絵をハメ込みたい場合に最適。
 - (●) **2.0x (2倍マージン - 標準・推奨)**: 被写体中心から 2 倍のワイド領域に拡大。被写体の輪郭から少しはみ出すような作画や影を描き足すのに最も扱いやすいサイズです。
 - () **3.0x (3倍マージン - 広域作画 / エフェクト)**: 被写体の外側に大きく飛び出すエフェクトやアクション線を描く場合に使用します。
 - () **カスタム倍率**: 任意の実数値 (例: 2.5 や 4.0) を直接指定できます。
 - ★ **お気に入り自動記憶**: 前回選んだ倍率は自動記憶され、次回も初期選択されます。

【B. Pin Bone Rig モード設定】

- **Target Rig Name (対象リグ名)** : 自動提案されます (例: Vegas_Rig_キャラ頭)。
 - 形状を歪ませたくないキャラクターやアイテム素材に対し、等倍基準の Pin Bone (ピンボーン) リグが自動作成されます。既存の作画レイヤーを保持したままアニメーションのみを安全に更新可能です。

【C. Stabilize Video モード設定】

- **Target Group Name (対象グループ名)** : Vegas_Stab_TakeName のように、テイク名に基づいたグループ名が自動提案されます。
- **Reference Frame (基準フレーム番号)** :
 - 被写体を「何フレーム目の位置・姿勢・サイズで静止させたいか」を指定します (デフォルトは 0 フレーム目)。
 - このフレームにおける被写体の位置と大きさが、Moho 画面のちょうど中央 (原点) に固定されます。

6. 実践ワークフロー & チュートリアル (Step-by-Step)

チュートリアル 1：看板・壁面への 3D パースハメ込み (Smart Warp Mesh)

実写映像の中にあるビル看板やポスターを、手描きのアニメ看板に差し替える手順です。

1. VEGAS 側:

- 看板の 4 隅を Mocha でトラッキングします。
- `Export_VegasMask_CSV.cs` を実行し、[Yes] `Perspective Mode` で書き出します。

2. Moho 側:

- `VegasMaskTracker` を実行し、書き出した `*_cornerpin*.csv` を選択します。
- [`Smart Warp Mesh`] を選択し、マージン倍率を `1.0x` (ジャストフィット) にして生成します。

3. 作画・合成:

- 自動生成された `Paint_Canvas` レイヤーに看板のイラストや文字を描き込みます (既存の画像ファイルをグループ内にドラッグ&ドロップして入れても OK です)。
- タイムラインを再生すると、カメラのパンや回転、パースの変化に合わせて、描いた看板が吸い付くように 3D 変形しながら追従します。

チュートリアル 2 : キャラクターへの小物・フキダシ 2D 追従 (Bone Rig)

歩いている人物の頭の上に「！」マークや吹き出し、名札を歪ませずに追従させる手順です。

1. VEGAS 側:

- 人物の顔や胸元をトラッキングし、ベジェマスクのトラックデータを作成します。
- `Export_VegasMask_CSV.cs` を実行し、[No] `Transform Only Mode` で書き出します。

2. Moho 側:

- `VegasMaskTracker` を実行し、書き出した `*_track*.csv` を選択します。
- [`Bone Rig`] を選択して生成します。

3. 作画・素材配置:

- 生成されたボーンの子レイヤー (`Paint_Canvas`) に吹き出しや漫符を描きます。
- 素材の形状は元の縦横比を保ったまま、人物の動きやスケール変化に合わせて綺麗に追従します。

チュートリアル 3 : 手ブレ実写動画へのアニメ的エフェクト作画 (Stabilize Video)

手持ちカメラで撮影した人物の周囲に、手描きのシャボン玉や魔法オーラを描き足すプロのワークフローです。

ステップ 1 : VEGAS で被写体をトラッキング

1. 揺れている手持ち動画クリップの被写体 (例: 人物の手や顔) を Mocha / ベジェマスクでトラッキングします。
2. `Export_VegasMask_CSV.cs` を実行し、CSV を書き出します。
 - **重要ポイント:** VEGAS 側の動画には手ブレ補正をかけず、**揺れているオリジナルのまま** Moho に読み込みます。

ステップ 2 : Moho でインバーススタビライズを実行

1. Moho のタイムラインに元動画を配置し、その動画レイヤーを選択します。
2. `VegasMaskTracker` ツールを起動し、CSV を選択して [`Stabilize Video`] を選びます。
3. 基準にしたいフレーム (例: 58フレーム目) を `Reference Frame` に指定して実行します。

4. **結果:** 動画レイヤー (**Video_Stabilizer**) に逆変換のアニメーションが付き、**激しく動いていた被写体が画面中央にピタリとフリーズ (静止)** します！

ステップ3：静止画感覚でエフェクトを描き込む

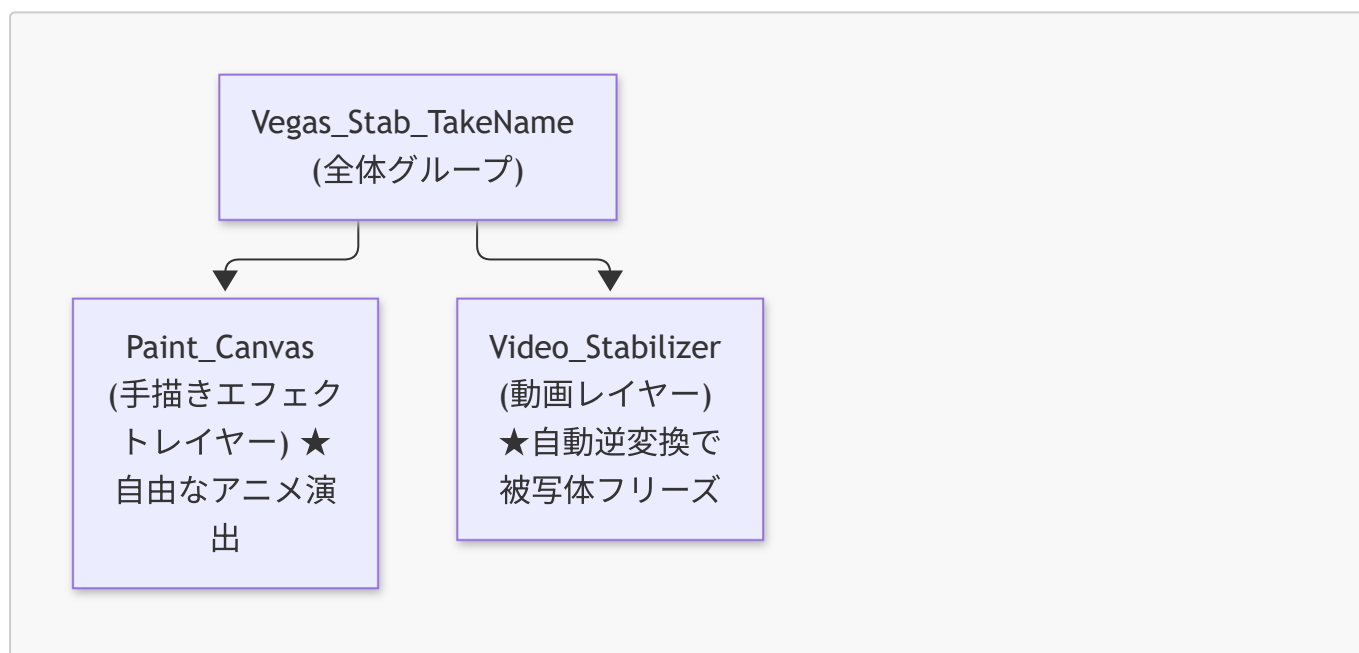
1. スクリプト完了時、上層のベクター画用紙 (**Paint_Canvas**) が自動でアクティブ選択されています。
2. タイムラインの各コマで、被写体の周りにシャボン玉やオーラ、雷光などを手描きします。
 - **圧倒的な描きやすさ:** 被写体が画面中央で止まっているため、手ブレに惑わされることなく、静止画の原画を描く感覚で驚くほど快適にペンを走らせることができます。

★ プロの演出極意：アニメ的誇張表現 (手動スケーリング・拡大) の加え方

実写動画では、被写体がカメラから離れると物理法則に従って小さくなります。しかしアニメーションの演出では：

「実写の被写体は遠ざかって小さくなっていくが、魔法のシャボン玉は逆にボワッと大きく膨らんで弾けさせたい！」

というような**アニメ的なデフォルメ (誇張表現)** が頻繁に求められます。本システムでは、この演出が極めて直感的に行えます：



- **手描きの独立性:** **Paint_Canvas** は動画レイヤーの「中 (子)」ではなく、独立した同階層に配置されているため、**動画の機械的な縮小に巻き込まれません。**
- **自由なキーフレーム演出:** 描いたシャボン玉の大きさや移動は、Moho のレイヤー変形ツールやポイント変形ツールを使って、**自分の意図するアニメーションのタイミング・サイズで自由に拡大・変形** させることができます。「物理的な静止はスクリプトに任せ、演出の誇張はクリエイターが自由に手付けする」という理想的な分業が実現します。

チュートリアル 4：実写の手の後ろをアニメキャラが通るオクルージョン合成 (画像シーケンス活用)

実写の人物の手の「後ろ側」にアニメキャラクターが隠れながら通過する高度な合成手順です。

1. VEGAS 側:

- Mocha で人物の手の形に沿ってスプラインマスクを作成します（4点ではなく、指の形に合わせた自由曲線で OK）。
- `Export_VegasMask_CSV.cs` を実行し、**[Yes] Perspective Mode** を選択。
- 次のダイアログで **[Yes] (Save PNG sequence)** を選択 します。手のシルエットの白マスク PNG 連番フォルダが書き出されます。

2. Moho 側でのレイヤー組み立て:

- Moho のレイヤーパレットで新規グループを作成し、グループのブレンドモードを **Masking: Hide all**（すべて隠す）に設定します。
- グループ内に以下の順序（上から下）でレイヤーを配置します：

```
[Mask Group] (Masking: Hide all)
├─ [Hand_Mask_Sequence] (書き出した白マスクPNG連番 / Masking: Subtract mask)
├─ [Anime_Character] (通過させたいアニメキャラや作画レイヤー)
└─ [Original_Video] (実写元動画 / Don't mask this layer)
```

3. 完成効果:

- アニメキャラクターが手の位置に来ると、手の形（指の1本1本）に合わせて自動的にキャラクターが切り抜かれて隠れ、「**実写の手の後ろを完璧に通り抜ける**」オクルージョン合成が **Moho 上で一発で成立** します！

!!IMPORTANT 【必読】作業画面（OpenGL プレビュー）でマスクが徐々にズレて見える現象と、その理由・対処法

Moho の編集画面でスペースキーを押してリアルタイム再生（またはタイムラインをスクラブ）すると、「**フレームが進むにつれてマスクと映像がだんだんズレていき、マスキングが追従していないように見える**」現象が発生します。これは動画ファイル（MP4 等）だけでなく、**元映像を連番画像（PNG / JPEG）にした場合であっても、作業画面上ではズレて表示されることがあります**。しかし、**Ctrl + R**（レンダリング）を押して確認するか、動画書き出し（Export Animation）を行えば、1ピクセルも狂うことなく完璧にマスキングされています。

なぜ作業画面（ビューポート）だけズレるのか？（Moho の OpenGL 表示の限界）

1. **超高負荷による OpenGL テクスチャ転送遅延:** 高解像度の画像シーケンス（マスク PNG 連番 + 元映像 PNG 連番）をリアルタイムに同時再生すると、毎秒数百 MB の画像データをディスクから読み込み GPU（VRAM）へ転送し続ける必要があります。Moho のリアルタイム再生エンジンはこの転送負荷に追いつけず、一方のレイヤーのテクスチャ更新が遅延・間引きされるため、再生が進むほど見た目のズレが累積していきます。
2. **OpenGL マスキングプレビュー（ステンシルバッファ）の仕様:** Moho のビューポートは軽快な動作を優先するため、簡易的な OpenGL ステンシルバッファでマスクをプレビュー表示しています。特に画像レイヤー同士のマスク処理では、テクスチャ座標の同期やキャッシュクリアがプロダクションレンダラーと完全一致しない既知の表示制限があります。
3. **レンダリング時は 100% 完璧に一致する理由:** **Ctrl + R** や最終書き出し時、Moho は時間を 1 フレームずつ完全に静止させ、ディスクから全レイヤーの該当フレーム画像をメモリ上に 100% 展開しきってから高品質な合成を行います。そのため、**トラッキングデータやマスキング設定自体は 1 ピクセルも狂わず完全に合致**しています。

作業時のおすすめ確認・対処ワークフロー

1. ★ 表示品質の「GPU Acceleration」を OFF にしてみる: Moho 画面右下の「Display Quality (表示品質)」アイコンをクリックし、GPU Acceleration のチェックを外すと、OpenGL のキャッシュズレが解消され、ビューポート上でも正しく表示されるようになる場合があります。
2. 連続再生ではなく「静止コマ + Ctrl + R」で確認する: 作業中はスペースキーでの連続再生に頼らず、タイムラインスライダーを止めて Ctrl + R (Render Frame) を押してください。0.5 秒で正確なマスキング合成結果をプレビューできます。
3. 動き全体を確認したい場合は「プレビュー動画作成 (Ctrl + Shift + R)」: アニメーションの動きを滑らかに確認したい場合は、メニューの ファイル → プレビューの作成 (Ctrl + Shift + R) を実行してください。低解像度の完全同期プレビュー動画が数十秒で生成されます。

7. Moho 内部レイヤー構造と非破壊作画保護システム

VegasMaskTracker が Moho 内に生成する階層構造は、長編アニメーション制作現場の過酷なリテイクにも耐えられるよう設計されています。

レイヤー階層ツリー図

```
[Vegas_Warp_看板A] (Smart Warp グループ または Stabilize グループ)
├── [Paint_Canvas] (ベクター作画レイヤー / 初期アクティブ)
│   └── ※ここに自由にイラストやエフェクトを描画！
├── [Warp_Mesh] (4頂点のメッシュレイヤー ※Smart Warp モード時)
└── [Video_Stabilizer] (元動画レイヤー ※Stabilize モード時)
```

制作を劇的に快適にする 5 大アシスト機能

① スキャッターブラシ等の即時作画保証（作画レイヤー自動選択）

スクリプト完了時、新規作成時だけでなく更新時（再インポート時）であっても、必ず子レイヤーのベクター作画レイヤー（Paint_Canvas）を特定してアクティブ選択状態にして終了します。親のボーンやグループが選択されたまま残ることがないため、インポート直後にスキャッターブラシや鉛筆ツールを持っても、PNG 保存画面が誤爆することなく、即座に快適に絵を描くことができます。

② 被写体フリーズ&静止画感覚お絵描き（インバーススタビライズ）

動画レイヤーに対し、トラッキングデータの位置・回転・スケールの逆行列を自動計算してバイクします。手持ちカメラの揺れを 100% 打ち消し、被写体をキャンバスの中央に固定します。

③ 階層セーフティ（グループ内実行クラッシュの完全防止）

Moho のレイヤーパレットで、どんな深いグループ階層やボーン階層を選択していても、スクリプトが自動的にルート階層を認識して安全に新規作成します。危険な階層移動による Moho 14 のクラッシュは一切発生しません。

④ 10倍スケール飛び & Frame 0/1変形ジャンプの完全撲滅

- **スケール飛び防止:** VEGAS の先頭フレームに存在する「未トラックの全画面マスク（幅 3.55）」と「実際の被写体（幅 0.35）」の段差（10:1）を物理的に自動検知し、トラック開始点の姿勢に完全クランプ。
- **メッシュ変形ジャンプ防止:** Frame 1 の追跡四角形頂点群（`frame1Pts`）を直接 Frame 0 のレストポーズ（`fPos`）およびキーフレーム（`fAnimPos`）に反映。
- **結果:** Frame 0 も Frame 1 もスケール 1.0000（等倍）で揃い、メッシュのひし形急変ジャンプも 100% 完全に消滅 します。

⑤ 完全非破壊の作画保護アップデート

既に `Paint_Canvas` やグループ内にたくさんのイラストを描き込んでいる場合でも、同名でスクリプトを再実行すれば、**絵には一切触れず、アニメーションキーフレームのみを最新のトラッキングに差し替え** ます。実写映像の再編集やトラッキングのリテイクがあっても、作画コストはゼロです。

8. 座標正規化と技術仕様（数学的背景と Fail-Safe）

画面解像度に依存しない「完全正規化パイプライン」

映像制作では、フルHD（1920x1080）、4K（3840x2160）、縦型ショート動画（1080x1920）など、多様な解像度が混在します。従来のツールでは、ピクセル単位の移動量をそのまま渡してしまうと、解像度が変わった瞬間にオブジェクトが画面外へ猛烈な勢いで吹き飛ぶ問題がありました。

VegasMaskTracker では、**VEGAS エクスポート側と Moho インポート側の双方で二重の正規化アーキテクチャ** を採用しています。

1. VEGAS エクスポーター（`Export_VegasMask_CSV.cs`）

プロジェクト設定から現在の正確な解像度（幅 W 、高さ H ）を動的に取得し、以下の数式で Moho の正規化座標系（中心原点、アスペクト比考慮）へ変換します：

$$X_{\text{moho}} = \left(\frac{X_{\text{pixel}}}{W} - 0.5 \right) \times \left(\frac{W}{H} \right)$$
$$Y_{\text{moho}} = - \left(\frac{Y_{\text{pixel}}}{H} - 0.5 \right)$$

これにより、画面中央が $(0, 0)$ となり、Y軸が上向き（Moho 準拠）のクリーンなデータが出力されます。

2. Moho インポーター（Dual-Defense フェイルセーフ）

万が一、古いバージョンのスクリプトで書き出されたピクセル値ベースの CSV が読み込まれた場合でも、Moho 側の `sanitizeRowsToNormalized` 機構が数値を検査（ $|Value| > 1.5$ をトリガーとして検知）し、プロジェクトの解像度に合わせて自動的に正規化を行います。いかなる環境下でもオブジェクトが画面外へ吹き飛ぶことはありません。

9. トラブルシューティング & よくある質問（FAQ）

本システムは厳格な **Fail-Fast 原則** に基づき設計されています。推測による適当な数値補完は行わず、問題が発生した場合は直ちに停止して詳細な原因と具体的な解決手順を報告します。

エラーメッセージ一覧と対処法

表示されるメッセージ	原因	対処法
No video event selected.	VEGAS タイムラインでクリップが選択されていない	対象のクリップをクリックして黄色い枠（選択状態）にしてから再実行してください。
No 'Bézier Masking' effect found...	クリップに対象エフェクトが適用されていない	クリップの Event FX に「Mocha VEGAS」または「ベジェマスキング」が適用されているか確認してください。
Mask region (white area) was not detected or is too small at frame X. (Detected: Y white pixels, Minimum required: 16+ pixels)	① Mocha の View Matte がオフ ② マスクが非表示/無効 ③ マスクが小さすぎる（15x15px未満） ④ 実写の影や暗部によるピクセル欠落	1. Mocha VEGAS のエフェクト設定で View Matte にチェック [X] が入っているか確認してください。 2. 該当フレームでマスクが有効になっているか確認してください。 3. 極小の被写体（15x15px未満）の場合は、エクスポート時に [No] Transform Only Mode （ピンボーン追従）を使用してください。 4. VEGAS標準ベジェマスキングの場合: 実写映像に直接適用するのではなく、「 単色 (白) 」イベントにマスクを適用して純白マットを作成してからエクスポートしてください。
Extracted mask quad at frame X is degenerate (width or height < 2px).	抽出された四角形の幅または高さが2ピクセル未満	マスクが潰れて直線化しています。Mocha 上で四角形が正しく面積を持っているか確認してください。
Position parameter not found for Transform mode.	エフェクト内に必要なトラッキングパラメータが見つからない	トラッキングキーフレームが正常に書き込まれているか確認してください。
No valid tracking data rows found in the CSV file.	CSV の中身が空、または不正な形式	VEGAS 側の書き出しが正常に完了した CSV であるか確認してください。

よくある質問（FAQ）

- **Q. VEGAS Pro 22 標準のモーショントラックやベジェマスキングを使うと、後半で座標が激しく暴れたりブレたりします。**
 - **A:** 「白い単色（Solid Color）イベント」を作成して、そこにマスクを適用して書き出してください。実写映像のままエクスポートすると、保存される画像シーケンスは白く見えていても、幾何解析エンジンが読み取る元画像は実写映像のままです。実写の影・服のシワ・暗い模様などが二値化判定で黒（マスク外）として切り落とされ、マスクの角が虫食い状に欠落して1フレームおきに数十ピクセル往復振動する原因になります。動画の上に「メディア ジェネレーター → 単色 → 白」イベントを配置し、そこにモーショントラックデータ（ベジェマスキング）を適

用してエクスポートすることで、実写ノイズが完全にゼロになり、Mochaと同等の完璧な吸着精度が得られます。

- **Q. Mocha でトラッキングした CornerPin が、Moho 上で小刻みに揺れたり暴れたりしませんか？**
 - **A: 全く暴れません！完全にマスクに吸着します。** 本システムの C# エクスポートエンジンは、1 ピクセル完全解像度走査と、幾何学的な**最大面積内接四角形アルゴリズム (MIQ: Maximum Inscribed Quadrilateral) **を採用しています。従来の間引きによる偶数グリッドの階段飛びや、局所的な頂点削除ノイズが完全に排除されているため、Mocha の白マスクの真の 4 コーナーに Moho の Smart Warp メッシュ頂点が 1 ピクセルの狂いもなく完全に吸着・一致します。
- **Q. 「Mask region was not detected or is too small」というエラーが出ました。**
 - **A: 以下の 3 点を確認してください：**
 1. **Mocha VEGAS の View Matte がオン [X] になっているか:** 最も多い原因です。VEGAS の Event FX 画面で **View Matte** にチェックを入れてプレビューが黒背景に白マスクになっていることを確認してください。VEGAS標準ベジェマスキングの場合は、白単色イベントに適用してください。
 2. **該当フレームでマスクが表示されているか:** トラッキング区間外でマスクが消えていないか確認してください。
 3. **被写体のサイズが小さすぎないか:** 4 隅メッシュ (CornerPin) の算出には最低 16 ピクセル (約 15x15 px 程度) の面積が必要です。極小の点や遠くの小物を追跡する場合は、エクスポート時に **[No] Transform Only Mode** (ピンボーン追従) を選択してください。
- **Q. 手ブレ補正 (Stabilize Video) を実行したのに、動画レイヤーがグループに入りません。**
 - **A: スクリプトを実行する前に、タイムライン上で動画レイヤー (または連番画像レイヤー) をマウスでクリックしてアクティブ選択状態 にしてください。** 選択されているレイヤーが自動的にスタビライザーグループ内に取り込まれます。
- **Q. 手ブレ補正でフリーズさせた後、描いたエフェクト (シャボン玉など) の大きさをアニメ的に拡大したいです。**
 - **A: Paint_Canvas** は動画レイヤーの縮小アニメーションとは独立しています。Moho の「レイヤー変形ツール」や各ポイントのスケールを使って、好きなコマで自由に拡大・縮小のアニメーションキーフレームを打ってください。
- **Q. 再インポートすると、せっかく描いたイラストが消えてしまいませんか？**
 - **A: 一切消えません。** 同名のグループが存在する場合、スクリプトは既存の **Paint_Canvas** や作画レイヤーには手を触れず、アニメーションキーフレームのみを最新のトラッキングデータに安全に差し替えます。
- **Q. 24 fps のアニメプロジェクトに、29.97 fps の実写トラッキング CSV を合わせられますか？**
 - **A: 完全に合わせられます。** インポート時の FPS Mismatch 警告ダイアログで **[Auto-Sync (Resample)]** を選択してください。CSV の実時間に基づいて自動リサンプリングされ、ズレなく完璧に同期します。
- **Q. 作業画面 (プレビュー再生中) で、画像シーケンスのマスクと映像がだんだんズレてしまいます (連番画像にしてもズレます)。**
 - **A: これは不具合ではなく、Moho の OpenGL 描画エンジンにおけるリアルタイム再生負荷 (複数画像シーケンスのテクスチャ転送遅延) およびステンシルバッファの表示制限による見かけ上のズレです。** **Ctrl + R** を押してレンダリングするか、アニメーション書き出しを行えば **100% 完璧にマスキングされています。** 作業画面上でのズレを改善したい場合は、画面右下の「Display Quality」で **GPU Acceleration** を **OFF** に設定するか、スペースキーでの連続再生ではなくタイムラインを一時停止して **Ctrl + R** で確認してください。全体の動きを確認したい場合は、**Ctrl + Shift + R** でプレビュー動画を書き出すのが最も確実です。

Copyright (C) 2026 Sentaku Clip. All Rights Reserved.