UNIVERSITY OF CRETE
SCHOOL OF SCIENCES AND ENGINEERING
COMPUTER SCIENCE DEPARTMENT

MotiVo Editor

A Human Motion Artistic Visualizer

Anastasia Rigaki

Thesis submitted in partial fulfillment of the requirements for the

Master's Degree in Computer Science and Engineering

Heraklion, June 2021

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.
This work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

UNIVERSITY OF CRETE

DEPARTMENT OF COMPUTER SCIENCE

MotiVo Editor

A Human Motion Artistic Visualizer

Thesis submitted by

Anastasia Rigaki

in partial fulfillment of the requirements for the

Master's Degree in Computer Science and Engineering

THESIS APPROVAL

APPROVED BY:

Author: Anastasia Rigaki

Supervisor: Constantine Stephanidis, Professor, University of Crete

---

Committee Member: Xenophon Zabulis, Principle Researcher, FORTH-ICS

Committee Member: Antonios Argyros, Professor, University of Crete

---

Director of Graduate Studies: Polyvios Pratikakis, Assistant Professor, University of Crete

Heraklion, June 2021

Acknowledgements

First of all, I would like to thank my Thesis supervisor and committee member, Professor Constantine Stephanidis for inspiring me to work in the field of Human-Computer Interaction as well as for his support throughout my Master studies. I am also grateful to my advisory committee members, Principal Researcher Xenophon Zabulis and Professor Antonios Argyros, who acknowledged my work by approving this Thesis. In addition, I would like to extend my appreciation to Nikolaos Partarakis for his insights and guidance by providing both technical and theoretical assistance. I would also like to thank Ilia Adami for her guidance in preparing and conducting the final evaluation experiment of the current work as well as for her valuable feedback during the design and development process.

Furthermore, I want to thank the Computer Science Department (CSD) and the Human Computer Interaction (HCI) Laboratory of the Institute of Computer Science (ICS) of the Foundation for Research and Technology - Hellas (FORTH) for equipping me with the means in order to achieve my research goals. I could not forget the Postgraduate Secretary of the Computer Science Department, Evangelia Kosma who had been always eager to help me and guide me about the postgraduate program.

I am also grateful to all my colleagues and friends and especially to Maria Doulgeraki and Eleni Tsolakou who encouraged and supported me throughout my Master studies.

Last but not least, I would like to express my deepest gratitude to my parents, Pelagia Savaki and Michalis Rigakis, my grandparents Georgios, Myron, Anastasia and Evangelia, who gave me their unconditional love and believed in me as well as they equipped me with the courage I needed to pursue my goals throughout my Master studies.

Part of the work reported in this Thesis has been conducted in the context of the European Commission H2020 Innovation Action Mingei (http://www.mingei-project.eu/) under grant agreement No. 822336.

Ευχαριστίες

Πρώτα απ’ όλα, θα ήθελα να ευχαριστήσω τον επόπτη της μεταπτυχιακής μου εργασίας και μέλος της επιτροπής, τον καθηγητή Κωνσταντίνο Στεφανίδη που με ενέπνευσε να εργαστώ στον τομέα της αλληλεπίδρασης ανθρώπου-υπολογιστή καθώς και για την υποστήριξή του κατά τη διάρκεια των μεταπτυχιακών σπουδών μου. Είμαι επίσης ευγνώμων στα μέλη της συμβουλευτικής επιτροπής μου, στον κύριο ερευνητή Ξενοφών Ζαμπούλη και στον καθηγητή Αντώνιο Αργυρό, που αναγνώρισαν το έργο μου εγκρίνοντας την παρούσα εργασία. Επιπλέον, θα ήθελα να εκφράσω την εκτίμησή μου στον Νικόλαο Παρταράκη για τις γνώσεις και την καθοδήγησή του καθώς παρείχε τεχνική και θεωρητική βοήθεια. Θα ήθελα επίσης να ευχαριστήσω την ‘Ιλια Αδάμη για την καθοδήγηση της στην προετοιμασία και τη διεξαγωγή του τελικού πειράματος αξιολόγησης της τρέχουσας εργασίας, καθώς και για τα πολύτιμα σχόλιά της κατά τη διαδικασία σχεδιασμού και ανάπτυξης.

Επιπλέον, θέλω να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών και το Εργαστήριο Ανθρώπινης Αλληλεπίδρασης Υπολογιστών του Ινστιτούτου Επιστήμης Υπολογιστών του Ιδρύματος Έρευνας και Τεχνολογίας - Hellas (ITE), για τον εξοπλισμό και τα μέσα που μου προσέφερε ώστε να πετύχω τους ερευνητικούς μου στόχους. Δεν θα μπορούσα να ξεχάσω τη Μεταπτυχιακή Γραμματέα του Τμήματος Επιστήμης Υπολογιστών, Ευαγγελία Κοσμά, που ήταν πάντα πρόθυμη να με βοηθήσει με το μεταπτυχιακό πρόγραμμα σπουδών.

Είμαι επίσης ευγνώμων σε όλους τους συναδέλφους και τους φίλους μου και ιδιαίτερα στη Μαρία Δουλγεράκη και την Ελένη Τσολάκου που με ενθάρρυναν και με υποστήριζαν καθ’ όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Τέλος, θα ήθελα να εκφράσω τη βαθύτατη ευγνωμοσύνη μου στους γονείς μου, την Πελαγία Σαββάκη και τον Μιχαήλ Ρηγάκη, στους παππούδες μου Γεώργιος, Μύρων καθώς και στις γιαγιάδες μου Αναστασία και Ευαγγελία, οι οποίοι μου έδωσαν την άνευ όρων αγάπη τους και πίστεψαν σε μένα, καθώς μου έδωσαν το θάρρος που χρειαζόμουν για να επιδιώξω τους στόχους μου καθ’ όλη τη διάρκεια των σπουδών μου.

Στους γονείς μου, Πελαγία και Μιχαήλ

MotiVo Editor: A Human Motion Artistic Visualizer

Abstract

2D motion visualization is an area that gains increasing interest as it provides an abstraction of motion scenes as well as summarization of motion activities. Artists, designers, and psychologists have explored techniques in which visual media can be organized to engender a narrative and direct viewer's attention within a scene. The need for such visualizations is currently used in many fields such as for illustrating a sport event in a newspaper, a dynamic experiment in a scientific journal or visualizing a craft process. Furthermore, 2D motion visuals can convey motion and instructions on printed matter and physical surfaces, making possible a wide range of applications that involve physical objects and surfaces (i.e. mixed reality).

Recently, such visualizations have been also explored in the domain of Digital Cultural Heritage (CH). Heritage Crafts (HCs) have cultural, historical, economical and societal significance and value and thus they are considered part of our Cultural Heritage. Nevertheless, HCs are threatened with extinction due to the declining number of practitioners and apprentices as well as the lack of interest in young age groups. In this context, motion visualization on static media could be considered valuable for the preservation of HCs for different purposes, including education, training and leisure.

Existing research works, on 2D motion visualization, are case-specific by exploiting one or two motion visualization techniques and they mostly address a narrow range of users (i.e. technology experts). Additionally, existing image editing software have extremely complicated user interfaces that require training and experience from the end-users.

To that end, this Thesis proposes an approach for motion visualizations inspired by techniques used in visual arts such as paintings, comics, photography and instruction manuals. We present MotiVo, a 2D Human Motion Artistic Visualizer (HMAV) editor that targets the preservation and representation of HCs, by assisting in the semi-automatic creation of readable visualizations and motion summarizations. The proposed system produces artistic image depictions of craft processes and techniques to enhance the educational and presentation value of digital content.

The key contributions of the MotiVo editor are: (1) it exploits multiple motion visualization techniques and creates artistic results; (2) it is user-friendly, as for the design of the system the minimum requirements were introduced; (3) it addresses a wide range of end-users (i.e. craftsmen, museum educators, researchers, academics, craft enthusiasts) and especially to novice users with no relevant experience in the image editing software community.

Keywords: 2D Motion Visualization · Image Processing · Computer Vision · Visual Arts

Πρόγραμμα επεξεργασίας MotiVo: Ένας καλλιτεχνικός οπτικοποιητής της ανθρώπινης κίνησης

Περίληψη

Η δισδιάστατη οπτικοποίηση κίνησης είναι ένας τομέας που αποκτά αυξανόμενο ενδιαφέρον καθώς παρέχει αφαίρεση των σκηνών κίνησης καθώς και σύνοψη δραστηριοτήτων που περιλαμβάνουν κίνηση. Καλλιτέχνες, σχεδιαστές και ψυχολόγοι έχουν διερευνήσει τεχνικές στις οποίες τα οπτικά μέσα μπορούν να οργανωθούν για να δώσουν μια αφήγηση αλλά και να κατευθύνουν την προσοχή των θεατών σε μια σκηνή. Η ανάγκη τέτοιων απεικονίσεων χρησιμοποιείται επί του παρόντος σε πολλούς τομείς, όπως για την απεικόνιση ενός αθλητικού γεγονότος σε μια εφημερίδα, ένα δυναμικό πείραμα σε ένα επιστημονικό περιοδικό ή την οπτικοποίηση μιας διαδικασίας τέχνης. Επιπλέον, τα δισδιάστατα κινούμενα οπτικά μπορούν να μεταφέρουν κίνηση και οδηγίες σε έντυπα υλικά και φυσικές επιφάνειες, καθιστώντας δυνατή μια ευρεία γκάμα εφαρμογών που αφορούν φυσικά αντικείμενα και επιφάνειες (π.χ. μικτή πραγματικότητα).

Πρόσφατα, τέτοιες απεικονίσεις έχουν διερευνηθεί επίσης στον τομέα της Ψηφιακής Πολιτιστικής Κληρονομιάς. Οι παραδοσιακές τέχνες έχουν πολιτιστική, ιστορική, οικονομική και κοινωνική σημασία και αξία και ως εκ τούτου θεωρούνται μέρος της Πολιτιστικής μας Κληρονομιάς. Παρ' όλα αυτά, απειλούνται με εξαφάνιση λόγω του μειούμενου αριθμού ασκούμενων και μαθητευόμενων καθώς και της έλλειψης ενδιαφέροντος από τις νεότερες γενιές. Σε αυτό το πλαίσιο, η οπτικοποίηση της κίνησης στα στατικά μέσα θα μπορούσε να θεωρηθεί πολύτιμη για τη διατήρηση των παραδοσιακών τεχνών για διαφορετικούς σκοπούς, συμπεριλαμβανομένης της εκπαίδευσης, της κατάρτισης και του ελεύθερου χρόνου.

Οι υπάρχουσες ερευνητικές εργασίες σχετικά με την οπτικοποίηση δισδιάστατης κίνησης, αφορούν συγκεκριμένες περιπτώσεις αξιοποιώντας μία ή δύο τεχνικές οπτικοποίησης της κίνησης και απευθύνονται σε ένα μικρό φάσμα χρηστών (π.χ. ειδικούς τεχνολογίας). Επιπλέον, τα υπάρχοντα λογισμικά επεξεργασίας εικόνας έχουν εξαιρετικά περίπλοκες διεπαφές χρηστών που απαιτούν εκπαίδευση και εμπειρία από τους τελικούς χρήστες.

Για το σκοπό αυτό, η παρούσα διατριβή προτείνει μια προσέγγιση για απεικονίσεις κίνησης εμπνευσμένες από τεχνικές που χρησιμοποιούνται στις εικαστικές τέχνες, όπως πίνακες ζωγραφικής, κόμικς, φωτογραφία και εγχειρίδια οδηγιών. Παρουσιάζουμε το MotiVo, έναν επεξεργαστή για τη δισδιάστατη αναπαράσταση της ανθρώπινης κίνησης με καλλιτεχνικό τρόπο, που στοχεύει στη διατήρηση και αναπαράσταση των παραδοσιακών τεχνών, βοηθώντας στην ημιαυτόματη δημιουργία ευανάγνωστων οπτικοποιήσεων και περιλήψεων κίνησης. Το προτεινόμενο σύστημα παράγει καλλιτεχνικές απεικονίσεις εικαστικών διαδικασιών και τεχνικών για την ενίσχυση της εκπαιδευτικής και παρουσίασης του ψηφιακού περιεχομένου. Οι βασικές συνεισφορές του επεξεργαστή MotiVo είναι: (1) εκμεταλλεύεται πολλαπλές τεχνικές οπτικοποίησης κινήσεων και δημιουργεί καλλιτεχνικά αποτελέσματα, (2) είναι φιλικό προς τον χρήστη, καθώς για το σχεδιασμό του συστήματος εισήχθησαν οι ελάχιστες απαιτήσεις, (3) απευθύνεται σε ένα μεγάλο εύρος τελικών χρηστών (π.χ. τεχνίτες, εκπαιδευτικοί μουσείων, ερευνητές, ακαδημαϊκοί, λάτρεις των τεχνών) και ειδικά σε αρχάριους χρήστες που δεν έχουν σχετική εμπειρία στην κοινότητα λογισμικού επεξεργασίας εικόνων.

Λέξεις-κλειδιά: Δισδιάστατη οπτικοποίηση κίνησης· Επεξεργασία Εικόνας · Υπολογιστική Όραση · Εικαστικές Τέχνες

Contents

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Motion is everywhere. Nature, everyday life, human activities are few examples interwoven with motion. One of Albert Einstein's well known quotes is, "Nothing happens until something moves''. Human perception of motion is mainly triggered by the visual system, to that end, throughout history, painters, sculptors, illustrators and directors have used multiple techniques in order to convey motion (e.g., colour contrast, motion lines, superimposition, and juxtaposition of visual frames). The depiction of motion is an important part of artistic expression as artists have depicted both motion and lack of motion as a way to stimulate interest [1].

Nowadays, motion visualization in static media provides an abstraction of motion scenes. As 'visual abstraction', we refer to a manipulation of realistic imaging (e.g., paintings, photographs) aiming to convey an understanding of the events occurring within a time-space interval. Motion can be effectively conveyed in static media via superimposed and juxtaposed images. Pertinent techniques are based on the cognitive capability of the observer to "fill-in" missing information. In this way, the depiction encodes an event taking place during a time interval rather than a photographic recording of a single moment. At the same time, applications of human motion visualization, such as illustrated safety or assembly instructions, still have a wide use in conventional depictions of human motion, namely 2D static depictions, whether these are presented on screen or on paper. Juxtaposition is used in illustrated instructions (e.g., manuals) as an ordered representation of images combined with superimposition of written information and graphical annotations, to direct the reader.

Up to now, these techniques were conventionally a manual task for illustrators, however, nowadays technology is offering a plethora of editing tools for digital creativity. It is common for graphic designers and illustrators to use image-processing software in order to simplify authoring and enhance visualization. Although these tools are a commodity, they still require insight and art skills from the illustrator.

Inspired by the depiction of human motion in the visual arts, we transfer pertinent visual approaches to the domain of human motion visualization in 2D. Our vision is to utilise these visualization techniques and create insightful visualizations of motion recordings on static media.

1.1 Human Perception

Humans due to their nature of being social creatures, are capable to perceive the context of an ongoing or future action based on their ability to infer from gestures and expressions. The perception of human action depends upon multiple sources of information including sensory, motor and affective processes [2]. This social skill of perceiving an action, intention or mood is greatly amplified by the human visual system which provides rich source of information [3]. Despite the fact that static pictures illustrate a person's affective state, motion provides even more vivid and reliable information. But what if we can combine them both? Psychologists and cognitive neuroscientists [4] have intensively studied the visual analysis of human action as well as phenomena of visual salience [5] [6] showing that outliers among visual features such as colour, size, and orientation preferentially attract one's attention. This attraction strength is modulated by multiple factors [5], including the scene itself and by the viewer's task. Visual techniques can further establish the order in which the eye visits elements in the scene. As Gestalt psychology [7] has studied, the human mind uses the perceptual grouping [8] in order to partition discrete elements into groups and form higher-order perceptual units [9]. This partition is based on some laws known as Gestalt laws (e.g. law of similarity, law of closure). These psychologists, as well as modern theorists, also suggested that there are certain "rules" that the visual system uses to determine which regions are the main figure, namely Figure-Ground Organization [10]. The idea of figure-ground organization is that, visual system structures the perceptual field into a figure and a background, relied on human brain interpretation of whole figures (e.g., objects), by separating them from the background and perceiving complex meanings through very simple and abstract elements. This procedure is called visual figure-ground segmentation [11] and it is considered as one of the first things humans do when they are looking at any composition.

Inspired by these theories of how humans infer from actions and figures in a scene, we aspire to create motion visualization results on static media.

1.2 Art & Motion Visualization

Before the twentieth century, traditional art was mostly oriented to still depictions of people and landscapes such as sculptures and paintings, nevertheless, due to the development of technology in the beginning of the twentieth century artists created more avant-garde artworks. The Futurism movement [12] brought alternative cues such as motion, speed and force that became part of the contemporary era including more abstract drawings and paintings. From all the art forms, Kinetic art is the most prominent on in terms of motion as it represents a reductionist approach, which emphasizes mostly on motion and de-emphasizes both form and colour [13].

Artists, designers, and psychologists have explored ways in which visual media can be organized to engender a narrative experience such as motion lines, colour contrast, juxtaposition or superimposition of one or more depictions. These techniques sequentially direct a viewer's attention and keep viewers oriented across transitions [14]. In comics and instruction manuals, juxtaposed images in deliberate sequences have been also used in order to convey motion based on the mental human visual process namely 'Visual Closure' [15] that correctly completes visual patterns even if they are not fully presented or presented in a segmented way. In photography and film recording, motion blur is a fundamental cue in the perception of objects in motion. This phenomenon is consisted by a sequence of frames and manifests as a visible trail along the trajectory of the object, resulting by the combination of relative motion and light integration taking place in film and electronic cameras. In such a photograph, given a stationary framework and a moving object, one can begin to grasp motion in the still image [16]. Besides photography, the effect of motion blur is also used in 2D computer graphics [17] [18]. In cinema, motion visualization techniques are also ubiquitous. Strong poses that fit the composition's objective, fractures of time and space as well as camera "tricks" are commonly used. This dissertation addresses the need for inspiration by such visualization techniques in order to produce artistic results as well as create motion narratives.

1.3 Research & Cultural Context

Human Motion Artistic Visualizer (HMAV) approach has been conducted in the context of the European Commission H2020 Innovation Action Mingei [19], which is exploring the possibilities of representing and making accessible both tangible and intangible aspects of craft as cultural heritage (CH). Heritage Crafts (HCs) involve tangible craft artifacts, materials, machines and tools, and encompass traditional craftsmanship as a form of Intangible Cultural Heritage (ICH). Intangible HC dimensions include dexterity, know-how, skilled use of tools, as well as identity and traditions of the communities in which craftsmanship is, or was, practiced [20]. The significance and urgency for the preservation of HCs is underscored by UNESCO, as "several are threatened with extinction, due to the declining numbers of practitioners and apprentices", in combination with demotivation, lack of interest in younger generations, and urbanization. Despite the cultural, societal, economic and traditional significance and value of HCs and ICH, efforts towards their digital representation, presentation, and preservation are scattered, both geographically and thematically. Although, mature artifacts and site digitization methods as well as repositories of corresponding digital assets are available, there are no approaches unifying the heterogeneous and multimodal knowledge on the representation of HCs. Mingei supports a wider vision of representation and preservation of HCs by implementing a variety of tools and applications that create the necessary conditions to not only preserve and support the transmission of HCs and associated Culture Heritage (CH) knowledge, but also to support scientific research in the domain of CH.

Visualization and illustration of human motion was selected by Mingei as a way to produce illustrated instructions on the performance of manual, possibly dexterous task that often involve the use of tools or machines. To that end, in the context of this Master Thesis, this approach was proposed in order to comprehensively visualize motion processes as well as exploit a plethora of artistic visualization techniques, supported by the MotiVo 2D editor. The 2D abstractions can provide insight in understanding the motion that the observer wishes to reproduce as well as convey motion and instructions on printed matter and physical surfaces, opening a wide avenue on applications that involve physical objects and surfaces (i.e., mixed reality).

The most significant novelty of our approach is that end-users with no relevant experience with image editing software would be able to create their visualizations out of video recordings, key poses and image depictions of craft processes and techniques to enhance the educational and presentation value of digital content.

1.4 Thesis Contribution

Inspired by the visualization of motion in art, comics, photography and instruction manuals, the goal of this Thesis is to build over centuries of visual art experience and move forward into proposing an approach for the artistic visualization of motion on static media. The need to display human activity in static media exists in many fields of modern life. Illustrating a sport event in a newspaper, a dynamic experiment in a scientific journal or visualizing a craft process are some examples of the need to display dynamic action in printed media.

This approach transfers artistic concepts in the digital world, inspired by human perception theory and known visualization techniques, in order to create insightful visualizations of human motion in the 2D world. The reason behind focusing on art is not sentimental, but practical. Artists have for long studied the perceptual appeal of such “tricks” over centuries of experimentation as the History of Art conveys [21]. Although there are techniques that offer a wide range of motion visualization tools, most of them focusing on Motion Capture (MoCap) [22] whereas other work exclusively on motion visualizations thus, there is still a need for a general approach that creates workflows between different visualization techniques and comprehensively visualizes motion on static media with artistic ways. Most of the existing works are case-specific or address to specific end-users familiar with technology, however, there has not been presented any approach that consists of interactive and simplified artefacts addressing to a wide range of end-users with no relevant experience.

In the context of this Thesis, we present the MotiVo editor, a 2D Human Motion Artistic Visualizer (HMAV) that targets on the preservation and representation of Heritage Crafts, by assisting in the semi-automatic creation of readable visualizations and movement summarizations as well as artistic image depictions of craft processes and techniques in order to enhance the educational and presentation value of digital content.

The MotiVo's editor novelty lies on the creation of hybrid depictions of motion and artistic visualization results, due to the facilitation of creating workflows and combinations between the offered functionalities. Moreover, a significant advantage of this editor is its simplicity and ease of use, as it was built on top of a plugin-based dockable layout which is familiar to the people usually using such applications and at the same time, it is simple enough to be used by non-experts. Regarding this, the MotiVo editor is highly extensible as if new functionalities arise they can be easily integrated.

1.5 Thesis Structure

In the following Chapters, the full extent of this Master Thesis will be presented. The remainder of this Thesis is divided into the following chapters:

- **Chapter 2** describes the background theory by which this Thesis was inspired, for the development of the MotiVo editor and its tools.

- **Chapter 3** presents the related work on 2D and 3D motion visualization techniques and tools.

- **Chapter 4** presents the design process of the MotiVo editor and examines the requirements that the MotiVo system met in order to fulfill its mission.

- **Chapter 5** provides an overview of the MotiVo editor and presents the design, functionalities as well as the implementation details of the integrated tools. Moreover, a complete use case is presented in which we have created two (2) versions of a comic book. Furthermore, out of the experience gained in this Thesis, several guidelines are offered for sufficiently visualizing motion using the MotiVo editor.

- **Chapter 6** describes the evaluation methods used during the design process of the MotiVo Editor, the heuristic evaluation process followed for the MotiVo editor as well as the final evaluation results.

- **Chapter 7** presents the conclusion of this Master Thesis as well as suggestions for future work.

- **Appendix A** Comic Book (Version 1)

- **Appendix B** Comic Book (Version 2)

Chapter 2

Background Theory

This chapter briefly presents the background theory by which this Thesis was inspired. The main objectives of this chapter are; human perception and motion visualization techniques used in visual arts (i.e. paintings, comics, photography and instruction manuals).

2.1 Theory of Human Perception

Human perception is the ability of recognition, organization and interpretation of sensory information and signals deriving from the environment [23]. In the context of this Thesis, we focused on the human ability to infer from the visual system. In the following subsequent sections, it is presented how the visual system is affected by the perceptual grouping [24] by using certain rules (i.e., Gestalt rules of grouping). Moreover, the 'Figure-ground' organization theory [25] is explained along with the way it determines which regions of a scene are considered as the main figures based on some visual properties.

2.1.1 Perceptual grouping & Gestalt laws of grouping

Perceptual grouping was first studied by Max Wertheimer, one of the fathers of Gestalt psychology [10] [26] [27] and refers to principles through which a set of discrete elements are partitioned into groups by the visual system, thus forming higher-order perceptual units [7] [24]. These perceptual grouping principles are powerful in the sense that humans usually cannot ignore them. Modern neurophysiology explains that human brains do not directly perceive objects and visual scenes but analyze each of them with respect to basic features [28]. Perceptual grouping occurs in all sensory modalities, such as vision [29], audition [30], somatosensation [31] and figure-ground segmentation [27] [10]. The earliest Gestalt work concerned perception, with particular emphasis on the visual perceptual organization as explained by the phenomenon of illusion [32], an optical illusion in which stationary objects shown in rapid succession, transcending the threshold at which they can be perceived separately, appear to move [33] [34].

Gestalt psychologists introduced the "Gestalt laws of grouping", principles that explain how humans perceive the world by eliminating complexity and unfamiliarity so that the mind would create meaning. These laws deal with the sensory modality of vision and they are referred in the literature as the law of similarity [35] [36], the law of proximity [37], the law of continuity, the law of closure [38], the law of symmetry [39], the law of past experience and finally the law of common fate [40] [10]. For instance, the law of similarity states that human perception is stimulated by seeing elements that physically resemble each other as part of the same object. This similarity can occur in the form of shape, colour, size, shading and other qualities and allows people to distinguish between adjacent and overlapping objects based on their visual texture and resemblance (Figure 1). The law of continuity states that elements arranged on a line or curve are perceived to be more related than elements not on the line or curve. In cases where there is an intersection between objects, humans tend to perceive the two objects as two single uninterrupted entities (Figure 2).

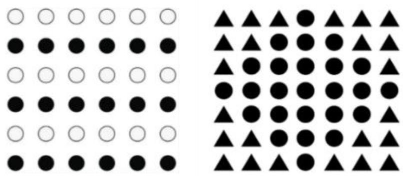


Figure 1. Gestalt law of similarity

Figure 2. Gestalt law of continuity

2.1.2 Figure–ground Organization

Figure-ground organization began by the Gestalt psychologists, most notably by Edgar Rubin [25], suggesting that there are certain "rules" that the visual system uses to determine which regions are the main figure [41]. Specifically, the idea of figure-ground organization is that it structures the perceptual field into a figure and a background based on human brain interpretation of whole figures (e.g., objects) by separating them from their background and perceiving complex meanings through very simple and abstract elements [42] [10].

This procedure is called ‘visual figure-ground segmentation’ and it is considered as one of the first things humans do when they are looking at any composition [11] (Figure 3). Figure-ground organization is a classical topic in the Gestalt psychology, and over the years many factors have been discovered that play a role in determining what regions are seen as figural. E.Rubin and the Gestalt psychologists identified some of the visual properties associated with figures known as *“configural cues”* [43].

Figure 3. Figure-ground perception

2.1.2.1 Figure-ground Segmentation

Figure-ground segmentation is a fundamental visual process in image processing and well-established in the Computer Vision literature. It is considered an internal part of the image segmentation field and focuses on two defined layers of an image, foreground and background. Specifically, figure-ground segmentation operates in an image scene by utilizing both static image cues (i.e., contrast, brightness, colour, texture) and temporal coherence cues (i.e., spatial model propagating figure/ground mask based on appearance, scale, spatial support) [44].

The importance of figure-ground segmentation was firstly recognized by the Gestalt psychologists. They distinguished figures and grounds in terms of their phenomenal appearance. According to Gestalt psychology, figures appear to have a definite shape whereas grounds are shapeless and behind figures. Nowadays, figure-ground segmentation is fundamental to the visual perception of objects and to visuomotor behaviour [11]. Many applications are based on figure-ground segmentation process such as image editing, object recognition, image retrieval, medical imaging or even recognition of human activities [45] (e.g., identifying handled objects and human motion).

2.1.2.2 Figure-ground Segmentation Techniques

Multiple techniques are available for figure-ground segmentation, each one relying on multiple criteria and producing results of different accuracy in segmentation as well. Every technique is used differently to images, as the selection of a specific class of an image is quite difficult in many cases. Up to now, there is not a universally accepted technique that can be applied in all cases of a figure-ground segmentation; every approach has advantages and disadvantages [46].

Bottom-Up Segmentation

The bottom-up segmentation approach is based on image cues (i.e., colour, texture) and search algorithms in order to find homogenous segments within an image. Specifically, in bottom-up segmentation, an image is first segmented into regions and following the continuity and similarity principles, pixels are grouped according to the grey/colour level or texture uniformity of image regions, as well as the smoothness and continuity of bounding contours (Figure 4). After the image segmentation, the image segments that correspond to a specific object are classified and grouped. Graph representation of an image (i.e., nodes represent pixels) and partition of the graph into subsets corresponding to salient image regions, is a common bottom-up approach [47]. Figure-ground algorithms based on bottom-up segmentation are more generic and they are called class-independent algorithms.

Top-Down Segmentation

Top-down segmentation relies on acquired class-specific information (i.e., prior knowledge about an object) and can only be applied to images from a specific class. This approach, uses object representation learned from examples to detect an object in a given input image and provide an approximation to its figure-ground segmentation. These include deformable templates, active shape models (ASM) and active contours [47] (Figure 4). Figure-ground algorithms based on top-down segmentation are categorized into class-specific algorithms.

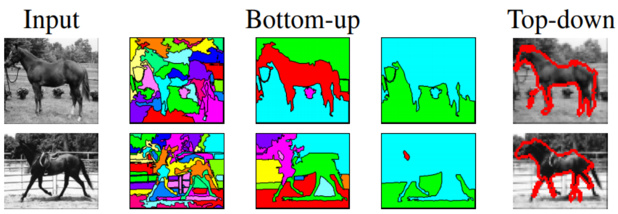


Figure 4. Bottom-up and Top-down segmentation [47]

Class-Independent Algorithms

The class-independent segmentation is a generic approach as it is carried out for any class of objects from the background. Two approaches are followed for class-independent segmentation. The first approach is based on multiple or hierarchical segmentation and the second one is exemplar-based segmentation transfer [45]. In the first one, by varying segmentation parameters a large set of regions is generated and then offline a ranking model is used to select regions likely to cover objects. This strategy exclusively depends on the bottom-up segmentation. In the second approach, exemplars are selected from the annotated training images which are geometrically similar to the query image. By merging segmentation masks of exemplars, the objects' locations are predicted for graph cuts to generate spatially consistent segmentations.

Class-Specific algorithms

In a class-specific segmentation approach, the input images should contain only a single class of objects. For each class of objects, the appropriate visual elements are extracted and used for objects of different classes to be distinguished. These class-specific features deliver a high amount of information such as shape and pose characteristics and expedite the segmentation process [48].

2.2 Motion in Visual Arts

Art before the twentieth century was mainly focused on landscapes and portrait paintings (e.g. nature scenes or stationary people standing or sitting still). The beginning of the twentieth century was a revolutionary time in the history of art, as modern artists for the first time tried to come up with new and often avant-garde ways (i.e. motion, speed) to capture images on a canvas [49] [9]. 'Visual movement' is the principle of art used to create the impression of action and is dependent on elements and principles of art such as rhythm, line, colour, balance and space. Movement can apply to a single component in a composition or to the whole composition at once. Nowadays, visual movement is not only used in paintings but also in many other visual art categories.

For this research work, we explored motion techniques used in art, comics, photography as well as instruction manuals. For each of the aforementioned domains, visual movement is adopted by different means.

2.2.1 Art and Painting

In art and paintings, many techniques are applied in order to create visual motion effects such as rhythm, texture, colour contrast, illusions, broken symmetry and affine shear. In this section, these techniques are briefly presented.

Rhythm

Visual rhythm can be used in art just like the rhythm used in music and it is expressed as lines, colours, and shapes. Rhythm is created when one or more elements of design are used repeatedly to create a feeling of organized movement and by purposefully arranging them with regard to pattern, harmony and variance. For rhythm to keep being exciting and active, variety is vital. The usage of a strong rhythm on paintings, directs the viewers' eyes around a scene as they move from one element to the next. There are many rhythms categories such as random, regular, alternating, flowing, or progressive. Artists often use varying types of rhythm within an individual work of art. For example, in Figure 5 the painting by Edvard Munch is an example of regular and flowing pattern rhythm.

Texture

Contrasting texture is a technique used in paintings to effectively convey motion. Specifically, by contrasting smooth and refined textures against thick and rough ones, depth is created and the illusion of movement directs the viewers' eyes as they are drawn towards these areas of rough and smooth paint. The famous artwork 'The Starry Night' (Figure 6) is an example in which different amounts of paint create texture contrasts and draw attention to specific areas.

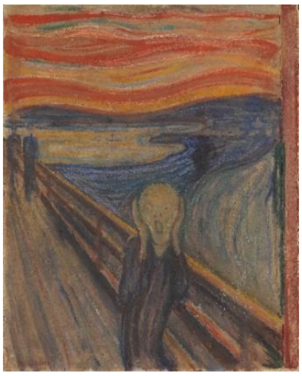


Figure 5. 'The Scream', Edvard Munch's 1893

Figure 6.'The Starry Night', Vincent van Gogh 1889

Colour Contrast

Dynamic use of colour can also enhance the feeling of visual movement just like dynamic lines do in still work of art [50]. The technique of colour contrast or broken color is expressed by using high and low key colours. High key colours have both high intensity and light values in contrast to low key colours demonstrated as more dark shades. The vibration between high and low key colours can effectively depict a sense of movement, as viewers' eyes tend to jump between all the different colors (Figure 7).

Illusions

Another concept of compositional movement is the "optical art" or illusion. Artists created movement using both repetition and contrast [51]. Optical art utilizes the human brain's interpretation efforts to organize the complex visual sensory information. Creating the sensory experience of movement in a still image, it engages and excites the viewer. For example, in Figure 8, the image feels like moving around and it is hard for the viewer to focus on a single spot.

Figure 7. ‘Haystacks’, Claude Monet 1891

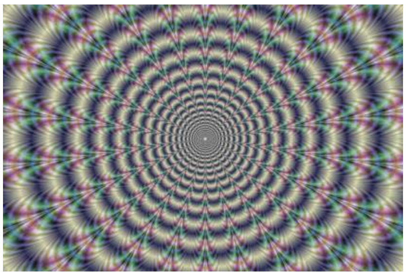


Figure 8. Visual motion - Illusions

Broken Symmetry (or Contrapposto)

Before the invention of photography, artists felt the need to show human figures in a more-or-less natural pose, thus to portray them, they created figures in dynamic balance or broken symmetry. Variable changes in body posture as well as balance can convey motion in a still work of art. This technique was called ‘*contrapposto’* [16] *and* was extensively used for the depiction of realistic human postures that would otherwise appear too static. In the technique of *contrapposto*, the human figure in paintings or sculptures was painted with hips and legs turned in a direction different from the shoulders and arms. Moreover, if hair and clothing blow in the wind, it includes dynamism to a person as an artistic object, however, the posture of that individual is still the most important feature with an elongated neck and torso as she stands in a classical contrapposto stance and her weight is shifted too far over the left leg for the pose to be held (Figure 9). Broken symmetry causes viewers to move their eyes more readily across a pattern.

Affine Shear

Affine shear is another technique that has been used by artists to represent visual motion throughout the 20th century and still is common today, specifically as seen in posters, comic and animated cartoons of popular culture. Artists depict the moving object as leaning in the direction of movement. This technique is even successful when that object is inanimate, such as a car or a train [16]. This lean is appropriated in the depiction of the motion of inanimate objects as objects take on the properties of a human being and, to initiate or to carry on with locomotion, it must lean forward as if to demonstrate the extra effort in overcoming inertia or in leaning against the wind. In the painting, Spanish Dancer, by John Singer Sargent (Figure 10), it is clearly illustrated the implied movement as the dancers head is well left of her feet and not centered over the feet. The relative orientation of the head and feet tells the viewer that she is in either motion or off balance. Artists can also imply movement by placing the subject element in space in a way that only makes sense if it is moving.

Figure 9. 'The Birth of Venus', Sandro Botticelli 1485–1486

Figure 10. 'Spanish Dancer', John Singer Sargent, 1882

2.2.2 Comics

Comics from their earlier days have faced the problem of representing motion in static mediums. Many cues can convey the impression of motion in a static comic image such as speed lines and visual closure through transition types. In the subsequent sections both techniques are described.

Motion or Speed Lines

Motion lines comprise one of many topics that have been studied, both by the history of art and neurosciences, due to (a) their efficacy in conveying information in intuitive ways for humans and (b) understanding how the human visual system works [52]. Speed lines [53] are a set of arrays containing lines attached to the moving object to define the trail of action but without deforming it. In mathematics, they are called vectors because they have two properties; direction and extent (i.e., speed) and were developed out of Greek parallelograms as representations of forces and motion. Motion or speed lines or zip-ribbons can be convenient as they denote the motion direction into a single static image [15] [54]. At first, motion lines (or zip ribbons) were irregular and messy but over the years, they became refined and even diagrammatic. Experiments [53] [55] have focused on the human brain's comprehension of motion lines, in cases of static panels that contain normal motion lines, anomalous lines or no motion lines at all. These studies have demonstrated that motion in panels, using normal lines, were viewed faster than when no lines were present, which were in turn perceived faster than in the case of anomalous lines. Moreover, it was found that a specific level of neural coding moving stimuli generates responses that are related to motion lines and human perception can be biased by motion lines due to their connection with the visual system when of viewers [52]. Illustration-inspired techniques such as speed lines and flow ribbons can be used in conjunction with animation to convey change over time. Bill Everett and Jack Kirby were the first to use motion lines in their hero's narratives as they were depicting action with drama. Motion lines can be drawn in many different ways according to the extent of the motion activity and speed that is taking place in a static medium. In the case of low-speed action, the moving object and the background are presented in a clear and static style while the motion lines are defined as a path of motion imposed over the scene (Figure 11). Another technique of motion lines with a high-speed effect is the one of a stable object that remains focused while the background is streaked [15] (Figure 12). This technique has been mainly adopted in Easter cultures such as Japan.

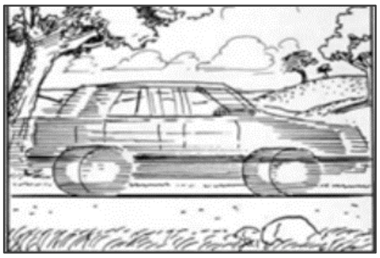


Figure 11. Motion lines in comics (low speed)

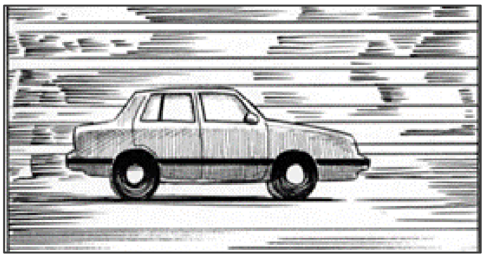


Figure 12. Motion lines in comics (high speed)

Inspired by the futuristic paintings, many comic artists used have strobe silhouettes to depict previous positions of the object (Figure 13). Similar to this, in the sixties and seventies, Marvel creator Gene Colan incorporated photographic streaking effects, which they were again multiple key poses but this time, not some realistic character depictions rather than sketched figures (e.g., shadows) that formed the action trail [15] (Figure 14).

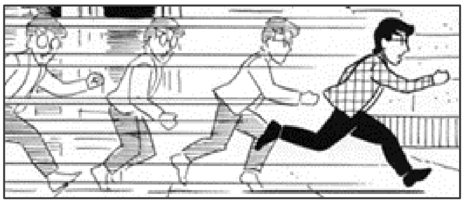


Figure 13. Motion lines in comics – Multiple key poses (realistic) of an action

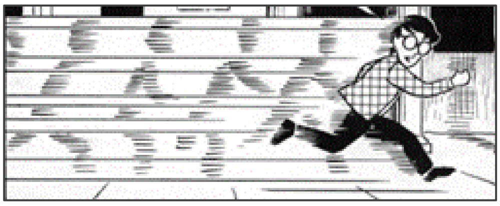


Figure 14. Motion lines in comics – Multiple key poses (sketched figures) of an action

Visual Closure and Transition Types

An alternative of conveying motion in comic narratives has been the sequential depiction of separate static panels each showing a partial act of a motion namely as juxtaposed illustrations. Visualizing motion as a sequence of juxtaposed key pose depictions provides a clear understanding of the illustrated motion. This technique is naturally perceived because of the mental human visual process namely Closure or Visual Closure [15]. Visual closure is a perceptual ability that allows humans to perceive correctly complete visual patterns even if they are not fully presented or presented in a segmented way. Human brains can perceive the "overall visual pattern" of the observed scene and detect, differentiate and understand the visual information by seeing little detail to recognize visual stimuli. These visual stimuli may be incomplete faces, figures, landscapes, letters, numbers, scenes etc. Visual closure is an especially important skill for children learning to read, since it is what enables a reader to recognize a sight word instead of having to sound it out each time. '*Closure'* as mentioned by McCloud, is extensively used for storytelling to provoke the feeling of suspense or challenge the audience. *Closure* is constant in electronic media such as filmmaking and television. Especially in comics, (visual) closure is the agent of change, time and motion and it is far from continuous. In comic panels, time and space are fractured creating a series of unconnected moments but it is the closure that allows human perception to connect these moments and create a continuous reality. Finally, *Closure* is significant for understating transitions between separate storytelling images in comics nevertheless; it has different impact according to the category of sequence transitions. There are several categories of sequence transitions in comics such as Moment-to-Moment, Action-to-Action, Subject-to-Subject, Scene-to-Scene, Aspect-to-Aspect and Non-Sequitur [15].

- **Moment-to-Moment (or slow-motion)**

  This transition is used in comics and gives a "slow-motion" feel by making each second an important one. It requires little closure because it uses the least amount of time (i.e., changes in panels take only seconds) (Figure 15 (a)) [15].

- **Action-to-Action**

  This transition is the most common type of transitions used in comics as it fully presents the transition steps of an action or scene. *Action-to-Action* transition often features single subjects in distinct frames. The first panel is commonly the beginning of an action and the last one is the end (Figure 15 (b)). Sometimes there are numerous panels in between depending on the complexity of the action [15].

- **Subject-to-Subject**

  This transition is taking place within a scene or idea. The degree of reader involvement is necessary to render these transitions meaningful. The time between the panels is simultaneous in many stories as well as space (Figure 15 (c)) [15].

- **Scene-to-Scene**

  This transition takes place across significant distances of time & space. A *Scene-to-Scene* transition is a powerful tool especially used to transport the audience to different times and places, with little to no explanation as to what has happened in between (Figure 15 (d)) [15].

- **Aspect-to-Aspect**

  In the *Aspect-to-Aspect* transition, time is completely either excluded or freeze and the representation of different aspects such as a place, an emotion, a mood of a scene are taking place For the most part in this transition "the wandering eye" activates the interpretation of an action by looking around and taking quick snaps to understand the context (Figure 15 (e)) [15].

- **Non-Sequitur**

  This transition offers irrelevant panels with no logical connection among them. This category relies on the idea of putting dissimilar images together and giving meaning and a relationship automatically (Figure 15 (f)) [15].

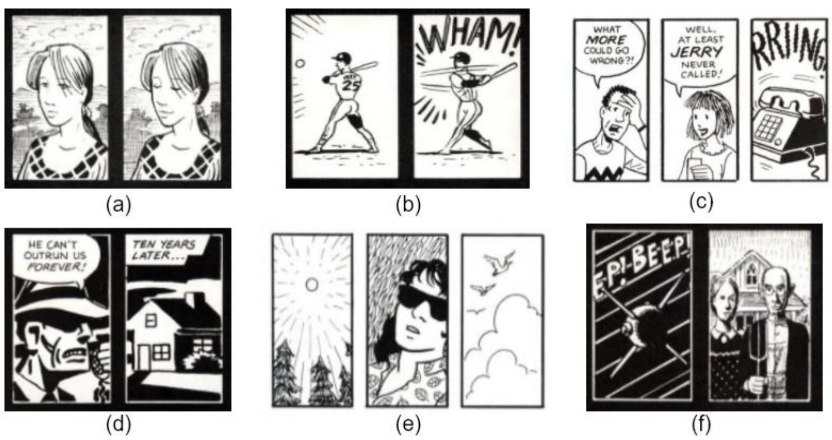


Figure 15. Visual Closure transition types (a) Moment-to-Moment, (b) Action-to-Action, (c) Subject-to-Subject, (d) Scene-to-Scene, (e) Aspect-to-Aspect, (f) Non-Sequitur [15]

2.2.3 Photography

Motion Blur

In photography and film recording, motion blur is a fundamental cue in the perception of objects in motion. This phenomenon is consisted of a sequence of frames and manifests as a visible trail along the trajectory of the object (Figure 16), resulting from the combination of relative motion and light integration taking place in film and electronic cameras (i.e., long-exposure photograph) [56]. Given a stationary framework and a moving object, one can begin to grasp motion in the still image (Figure 17). In general, sequences containing a moderate amount of motion blur are perceived as natural, whereas its total absence produces jerky and strobing movement.

In addition, background/foreground extraction techniques have been used in photography to transfer the point of interest of the viewer from the background to foreground and vice versa. This is something also called by visual artists and photographers as *'selective focus'*. Besides photography, the effect of motion blur is also used in 2D computer graphics [17] [18].

The simulation of motion blur is vital for both animated sequences and still images among the objects on a scene. In the last decades, many graphic experts have developed tools for representing motion via filters applied on static media such as software products (e.g. GIMP, Adobe Photoshop). Although the motion blur technique is broadly used for motion visualization in static media, a potential consequence is the misinterpretation of the motion direction. In such cases, viewers are not able to apprehend the start and the end of the depicted motion and thus the overall chronological order of an action in a scene.

Figure 16. Motion blur in photography – trail along the trajectory of dice

Figure 17. Motion blur in photography – Selective focus

Multiple images

Representation of motion and activity in longer time intervals or scenes has been treated in photography by manipulating time in the depiction, so that multiple time instances are seamlessly summarized, or “gracefully superimposed” without affecting the realism of depiction. Multiple images is a technique broadly used by many artists and can be found at least two to three hundred years before the advent of photography [16]. Unlike a single image, a series of discrete static images can render the impression of motion without sacrificing the resolution or visual clarity of the moving object (Figure 18) [57]. The *‘multiple-exposure'* technique can depict motion in still images and paintings by creating a tempo of successive poses of a figure denoting either the direction or the object's movements across a scene (Figure 19).

Figure 18.Man walking, Marey 1891

Figure 19. Cheval blanc monté, Marey 1886

2.2.4 Instruction Manuals

In instruction manuals and specifically on assembly instructions and user guides, two (2) are the main techniques used for motion visualization; juxtaposition and superimposition.

Juxtaposition

Juxtaposition in illustrated instructions is used as an ordered representation of images in deliberate sequences depicting a different time fragment of a motion action. The principle is that such depictions utilize the “*visual interpolation*” [42] or “filling-in”, in the temporal dimension [58]. Visualizing motion, as a sequence of juxtaposed key pose depictions, provides a clear understanding of the illustrated motion [58].

Superimposition

Superimposition is the placement of one thing over another so that both are still evident. In instruction manuals, superimposition in the form of annotations (i.e. arrows, lines, numbers)

[16] provides visual clues of motion on static scenes that direct the readers as it assists the understanding of visual interpolation (Figure 20).

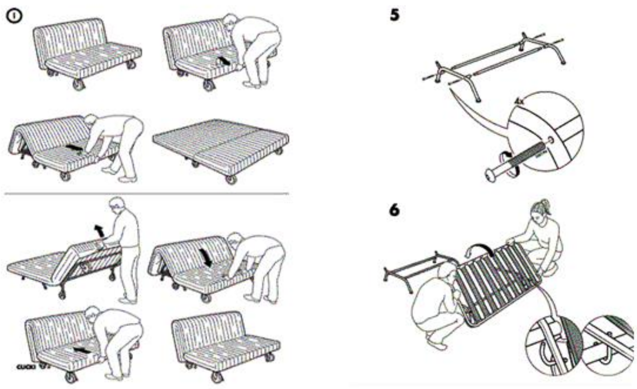


Figure 20. Motion visualization in instruction manuals

Chapter 3

Related Work

Several research approaches have studied motion visualization both in 2D and the 3D world as well as developing tools that generate motion visualization results. The majority of these approaches are mainly focused on 3D visualization techniques using motion capture data etc., thus 2D visualization techniques are still at an early stage, however they are rising nowadays. This chapter, presents the literature review of the best known 2D and 3D techniques and tools.

3.1 2D Motion Visualization Techniques & Tools

Over time, due to technical state-of-the-art alternatives of human motion digitation, a large number of existing studies in the broader literature have examined 2D motion visualization techniques [59]. The technique of '*action summarization*' is prevalent in the motion visualization community, as motion effects can be produced by processing a series of animated scenes and extracting still image frames as a result. Moreover, a wide range of motion effects such as motion lines or motion blur has been adopted in 2D current approaches.

Action Synopsis [60] is an approach that as input automatically selects poses taken from an animation sequence (3D) or a video clip (2D) and presents motion in still images. A similar work is Action Snapshot [61] which is a method based on information theory that automates the process of generating a narrative still image by obtaining the optical key poses of animated scene sequences. In addition, in Motion belts [62] motion data is visualized by unfolding a motion into 2D stripes of key frames on a timeline. Comparing these approaches with ours, they mostly lack in terms of variety on motion visualization techniques as they focus mostly on the technique of multiple images.

Simon Bouvier-Zappa et al introduced a system [63] that generates artist-inspired motion cues (i.e., motion arrows, multiple key poses) and synthesizes a single 2D image of an animated character derived from 3D skeletal motion capture data. Another research work [64] introduced by Okihide Teramoto creates motion photographs. Specifically, given a still image with a moving object, users can add or edit multiple motion effects. This approach is similar to us, however, it lacks in terms of our artistic results as we do not only generate motion images but we can convey motion in static mediums through many vivid artistic techniques.

This approach [65] presents a 2D-based post-process method that creates motion photographs from a single image using the motion visualization techniques of speed lines, arrows repetitive contours, motion blur etc. In the interface [66] introduced by M. Choi et al. users can browse overall motion by viewing 2D stick figures as unified images generated from the database and retrieve them directly by using sketched stick figures as an input query. This approach is far from ours as it uses motion lines to produce sketched images.

Some research approaches have also studied motion visualization on cartoons or even have been inspired by them. For instance, a technique [67] introduced by Christoph Bregler et al., captures the motion style from traditionally animated cartoons and retarget the same style to a different domain such as 3D models, 2D drawings, and photographs. Moreover, another 2D technique is [68] which makes fast-moving objects appear blurred and emphasizes the motion of cartoon objects through the application of geometry. A similar work [69] is a system that generates images of 3D scenes that depict dynamics following traditional design principles of visual arts and storyboards.

Another research work [70] is an image processing method that uses steerable filters to change the phase of image pixels. When the phase is continuously changed, a static image appears to move in a specified direction. However, the original colour intensity of the input image is hard to maintain. This computational method [71] generates self-animating images from given input by using a visual illusion phenomenon called a rotating snake. Users can perceive the flow of a motion field even though the pattern itself is static. Nevertheless, it is hard to control the speed of motion and even impossible to simulate the illusion while maintaining the original pixel colour.

The aMotion Toolkit [72] allows users to create motion textures as brushes whose effects can be applied to a surface (an image or a video). Byungmoon Kim et al., have proposed a semi-automatic approach [73] for adding expressive renderings to images and videos that highlight motions and movement by utilizing a variety of motion effects, such as a time-lapse ghost, a temporal flare, a particle effect and speed-lines.

This system [74] is capable of rendering motion within a video sequence in artistic styles such as cartoon-style motion cues, augmentation and deformation cues. The main difference of all the previous approaches to our work is that they used a video sequence as an input and offered no support for interactive user annotations for image input. Lastly, an ongoing research work namely MotionNote [75] includes MoCap and reconstruction on an avatar, motion visualization on a unit sphere and motion notation on a 2D EPG.

3.2 3D Motion Visualization Techniques & Tools

Regarding 3D Motion Visualization in virtual reality, a wide range of literature contributions have been proposed to different domains such as sports, medicine and art. Key Probe [76] is a key-frame extraction technique, relied on an algorithm appropriate for rigid-body and soft-body animations that converts a skeleton-based motion or animated mesh to a key frame-based representation. Based on motion depiction techniques in existing 3D motion sculptures and 2D comic art, ChronoFab [77], is a 3D modeling tool that enables users to visualize motion with static and transient visuals (i.e., motion lines, multiple stroboscopic stamps, sweeps, particle systems) using time and dynamic simulation.

In this research work [78] there are presented real-times methods for cartoon shading and pencil-sketch rendering of real-time animation as well as techniques for emphasizing the motion of cartoon objects by introducing geometry into the cartoon scene. Similar work has also been developed for artistic motion visualization. Yang introduced a framework [79] that produces rough drawings of images that tend to emphasize thicker strokes for the most important features. Moreover, this approach [80] is focused on motion visualization in computer games and utilizes three rendering motion techniques such as squash-and-stretch, multiple images and motion lines.

MoSculp [81] is a system that requires a video as input and visualizes human motion via 3D motion sculptures. Specifically, users can customize the sculpture design, for example, (e.g., materials, lighting conditions) and explore the sculpture directly in 3D or physically print it. Another work for authoring illustrations of human movements introduced by Pei-Yu is DemoDraw [82] a multi-modal system that translates speech and 3D joint motion into a segmented sequence of key poses and salient joint movements. Based on these sequences, it automatically generates a series of motion illustrations in effective and understood illustration styles. 3D visualization has also been proven valuable in the demonstration of Motion Capture (MoCap) data. Such an example is TooltY [83], a 3D authoring platform for the demonstration of simple tool operations (e.g. usage of a hammer, scissors) in 3D environments.

Motion Doodles [84] is also an interface that uses a 3D cursive language for sketching either 2D or 3D character motion. Johannes Schmid et.al, from Disney Research, has proposed a 4D data structure [85] that aggregates an object's movement into a single geometric representation by sampling an object's position at different time instances and connecting corresponding edges in two adjacent samples with a bilinear patch. Motion visualization of the objects is achieved using a set of programmable motion effects for various styles (e.g., speed lines, multiple stroboscopic images, temporal offsetting, stylized blurring etc.) on both simple and production examples.

In this research work [86], is introduced a system for real-time tracking and visualization of body motion in a semi-immersive virtual environment; firstly real-time tracking and rendering are taking place to display one or more user movements in a single virtual scene, then after capturing the desired motion, the users can further customize the visualization of individual motion trails. In the domain of medicine there are many literature approaches. An example of that is a non-invasive method for diagnosis of sleep apnea [87] useful in sports medicine and YOGA practice based on respiratory motion visualization taking a 3D movie using a Time of Flight (ToF) camera and extracts the respiratory motion, which is analyzed and displayed in real-time. In addition, this platform [88] has been designed to visualize the 3D motion of the human tongue, using ultrasound image sequences and modal analysis to perform the simulation and understand speed production in real-time. There is also a model [89] that visualizes and analyzes the motion of the left ventricle from a time series of 3-D images.

Depth information of animations assists summarization of 3D animations in a single image. Specifically, there has been proposed a method [90] that extracts important frames from the animation sequence based on the importance of each frame by its amount of contribution to the gradient and constructs a composite depth image and its gradient image.

In the domain of sports, plenty of motion visualization methods are introduced. For instance, Yijun Shen et.al, have introduced a method [91] that visualizes the high-level skills of boxers using automatic motion analysis and visualization framework to evaluate high-level skills such as the richness of actions, the flexibility of transitions and the unpredictability of action patterns. Lucent Vision [92] is a visualization system that has been developed for the sport of tennis. It uses real-time video analysis, extracts motion trajectories and provides a variety of visualization options. Moreover, there has been an analysis and visualization framework [93] for swimming motion that uses virtual reality to display 3-dimensional models of swimmers by digitizing their motions and creating personalized virtual representations.

3.3 Progress Beyond the State-of-the-Art

By reviewing the related work and state of the art, it appears that, until now, most of the research works have focused on 3D motion visualization in contrast to 2D motion visualization. The available 2D motion visualization approaches are not generic, but case-specific and exploit a few motion visualization techniques without considering concepts of artistic representations. Moreover, most of them lack in terms of intuition and ease of use. In case of the available image editing software (e.g., Photoshop, Gimp), although they support a plethora of functionality for motion visualizations, they are complicated software suites that require extensive training and insight skills from novice users.

In the context of this research work, we have identified a gap between motion digitization and artistic motion visualization. This Thesis proposes an approach for bridging these dimensions into a single workflow by building over centuries of visual art experience and exploring motion visualization techniques on static media with artistic ways. The proposed system is the MotiVo editor, a 2D Human Motion Artistic Visualizer that targets the preservation and representation of Heritage Crafts and assists in the semi-automatic creation of static visualizations as well as motion summarizations in order to enhance the educational and presentation value of digital content. The MotiVo editor was inspired by motion visualization techniques used in the visual arts (i.e. paintings, photography, comics) as well as by the studied background theory (i.e. Gestalt laws of grouping, Figure-ground organization). Each of the selected visualization technique used is approached by the MotiVo editor as a different image editing tool. For instance, inspired by the theory of figure-ground organization, the MotiVo editor enables the users to create foreground/background image masks for the amplification of foreground scenes in order to illustrate craft practices. Moreover, inspired by the Gestalt law of 'similarity' and 'continuity', the MotiVo editor, in terms of colour, takes into account that the human perceptual system is sensitive to contrast changes and enables users to give more contrast to areas and objects of interest. At the same time, in the case of visual movement summarization, the proposed system gives the essence of movement through the interpolation of static frames as well as the visualization of key point trajectories.

A noteworthy aspect of MotiVo editor is that apart from exploiting various motion visualization techniques, it deploys workflows between the available techniques and thus, users can create artistic hybrid results for the representation of HCs. Compared to the reviewed literature approaches, the MotiVo editor addresses a wide range of users (e.g. craftsmen, museum curators and exhibitors) with the intention to narrow down the gap between people that have expertise in advanced technology and tools and people that are considered CH stakeholders. The strategy behind the implementation design of the MotiVo, is the simplicity and ease of use and thus it addresses users with no relevant experience on image editing software and requires minimum expertise and knowledge from the user side. Finally, the MotiVo editor is highly extensible as if new artefacts arise they can be easily integrated.

Chapter 4

Design Process

An iterative design process was followed throughout the development lifecycle of the MotiVo editor, according to the principles of User-Centered Design (UCD) [94] that were also adopted by the Mingei H2020 EU project. UCD [95] is an iterative design process for interactive applications, systems, and products. Its main characteristic is that it places the end-users and other identified stakeholders' needs at the center of each design and development phase of the system (tool, application, or product). The main goal of this process is to ensure that the resulted system meets the user's needs, supports their goals and objectives, and satisfies the main parameters of usability: ease of use, learnability, effectiveness, efficiency, and satisfaction. There are four main stages in the UCD process (Figure 21):

1. Understand the context of use
2. Specify user requirements
3. Design solutions
4. Evaluate against requirements

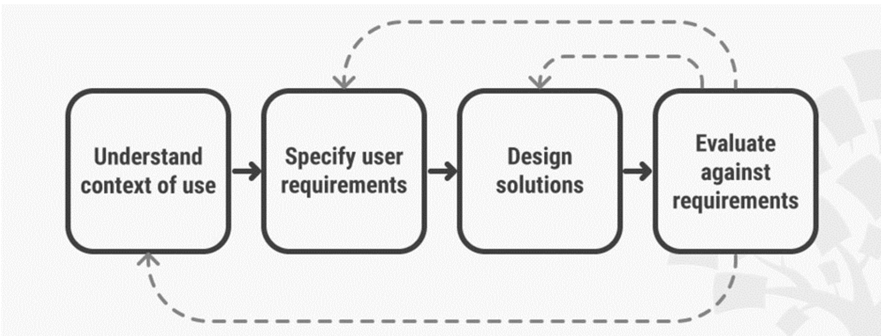


Figure 21. The four stages of the UCD process.

4.1 Understand and specify context of use

This stage involved the understanding of the context of use of the MotiVo editor as well as the end-users and other stakeholders, their goals and expectations from using the system and the environment in which the system will be implemented.

- **Description of context and rationale**

  The artistic visualization of motion is used for many decades as a means of insightful demonstration of manual tasks and tools usage. Know-how and expertise stems from cartoons, illustrated instructions, computer animation & graphics etc. These requirements regard the need for artistic summarization of human motion, particularly for non-interactive presentations of crafts, such as in books, flyers, educational material, instruction manuals. To that end, the Human Motion Artistic Visualizer approach addresses not only to the problem of maintaining craft processes via different means of visualizations by also contribute to new means of motion visualization in static media. In this context, the MotiVo editor was developed.

- **Stakeholders mapping**

  Craftsmen, Museums & CH institutions, Curators, Museum educators, Content owners & Creative Industries, Researchers and Academics, Craft enthusiasts.

4.2 Specify requirements

In this stage, the requirements the MotiVo editor were specified to satisfy the users' needs and fulfil its objectives. These requirements were based on the need for alternative ways of artistic human motion visualization, particularly for non-interactive presentations of crafts. The main requirements for the MotiVo editor are presented in Table 1.

Table 1. MotiVo editor – main requirements

Requirement 1	The system should support the creation of a new project with specific file extension (.hmav)
Requirement 2	The system should support the saving of the progress achieved in a created project
Requirement 3	The system should support the loading of a saved project
Requirement 4	The system should load the visualization sources (images, 2D trajectory files)
Requirement 5	The system should support image based averaged motion visualization.
Requirement 6	The system should support image based weighted motion visualization
Requirement 7	The system should support artistic visualization of trajectories within image files.

Requirement 8	The system should support artistic visualizations superimposed upon image sources
Requirement 9	The system should support visualization based on juxtaposed illustrations (cartoons).
Requirement 10	The system should support the isolation of a foreground scene.
Requirement 11	The system should support the creation of comic pages.
Requirement 12	The system should support the exportation of the visualization results in multiple output formats.

The requirements for MotiVo editor are extensively presented in the next section (Section 4.2.1).

4.2.1 The MotiVo's Editor Requirements

Requirement 1. The system should support the creation of a new project with specific file extension (.hmav).

Use Case Title	Create Project
Brief description	Users can create a new HMAV project
System input	The user inserts the details of the new project (i.e., name, images folder path, trajectories folder path)
System output	A new project representation in the user's file system with the .hmav extension
Scenario steps	Step 1. The user starts the HMAV Step 2. The user provides project name in the new project dialog Step 3. The user provides project folder path by browsing the file system Step 4. The user provides the file source of image assets Step 5. The user provides the file source of trajectories file(s) (optional) Step 6. The user selects to create the new project

Requirement 2. The system should support the saving of the progress achieved in a created project.

Use Case Title	Save Project
Brief description	Users can save the progress on a currently loaded HMAV project
System input	The user makes changes and decides to save a project
System output	The system saves the project progress to the project's path and project files
Scenario steps	Step 1. The user makes some progress on a loaded or created project Step 2. The user selects to save the progress of the currently active project using the save button which located on the main window

Requirement 3. The system should support the loading of a saved project.

Use Case Title	Open project
Brief description	Users can load a pre-existing HMAV project
System input	The user provides the path of a pre-existing HMAV project
System output	The system loads source images, trajectories and visualizations from a pre-existing project
Scenario steps	Step 1. The user starts HMAV editor Step 2. The user locates the open project functionality at the top bar of the HMAV Step 3. The user locates the ".hmav" project file using the file system browsing window Step 4. The user selects "open" to load the pre-existing project

Requirement 4. The system should the loading of visualization sources (images, 2D trajectory files).

Use Case Title	Load visualization source (images, 2D trajectory files)
Brief description	Users can add new sources to an HMAV project
System input	A number of new sources to be added to a project
System output	The sources are added to the project and the project file gets updated
Scenario steps	Step 1. The user initiates the "add new" functionality Step 2. The user selects the type of source to be added Step 3. The user locates the sources to be added

	Step 4.The user adds these sources to the active HMAV project

Requirement 5. The system should support image-based averaged motion visualization.

Use Case Title	Image based on averaged motion visualization
Brief description	Users can perform an averaged motion visualization based on a collection of frames depicting several key-steps of the motion to be visualized
System input	A selection of image frames for performing a averaged motion visualization
System output	The result of the application of the averaged visualization algorithm to the source "key frames"
Scenario steps	Step 1. The user selects the visualization tool Step 2. The user defines the source frames to be used by the visualization tool Step 3. The user initiates the visualization functionality Step 4. The user previews the visualization outcome

Requirement 6. The system should support image-based weighted motion visualization.

Use Case Title	Image based on weighted motion visualization
Brief description	Users can perform an image based weighted motion visualization based on a collection of frames depicting several key-steps of the motion to be visualized
System input	A selection of frames for performing a weighted motion visualization and a set of weights for each key frame
System output	The result of the application of the weighted motion visualization algorithm to the source key frames and using the provided set of weights
Scenario steps	Step 1. The use selects the visualization tool Step 2. The user defines the source frames to be used by the visualization tool Step 3. The user defines the weights to be used for each frame Step 4. The user previews on runtime the visualization outcome

Requirement 7. The system should support artistic visualization of trajectories within image files.

Use Case Title	Artistic visualization of trajectories within images
Brief description	Users can visualize the trajectories of a specific joint of a user's body using an image source as the background of the visualization
System input	An image and a set of trajectories together with the visualization style
System output	The result of the application of the visualization style for the selected trajectories of the weighted motion visualization algorithm to the source "key frames" and using the provided set of weights
Scenario steps	Step 1. The user selects the appropriate tab corresponding to the artistic visualization of trajectories within images Step 2. The user defines the source image to be used for the background of the visualization Step 3. The user defines the source trajectories to be visualized Step 4. The user previews on runtime the visualization result Step 5. The user produces and saves the visualization

Requirement 8. The system should support comic/artistic visualizations superimposed upon image sources.

Use Case Title	Comic/artistic visualization superimposed upon the image source
Brief description	Users can perform a manual visualization of motion within an image
System input	An image source and one or multiple ready to use concepts and icon sets (i.e., arrows, lines, etc.), such as those in comics
System output	The result of the application of the visualization concepts and icons and their attachment to the source image
Scenario steps	Step 1. The user selects the source image Step 2. The user selects the type of visualization and places it in the desired position within the image Step 3.s The user selects the desired appearance effect for the selected visualization Step 4. The user previews the visualization Step 5. The user produces the visualization result when he is satisfied by the preview

Requirement 9. The system should support visualization based on juxtaposed illustrations (cartoons).

Use Case Title	The system should support visualization based on juxtaposed illustrations (cartoons)
Brief description	Users can create juxtaposed illustrations
System input	A selection of frames for performing a juxtaposed illustration, illustration filter for each frame and collection of manual annotations of each frame
System output	The result of the application of the illustration filter and manual annotation to the selected source image frames
Scenario steps	Step 1. The user selects the visualization tool Step 2. The user loads the selection of images to be used as sources of the juxtaposed visualizations Step 3. The user edits each image by selecting illustration type Step 4. The user inserts manually dialogs (optional) in each frame of the juxtaposed illustration Step 5. The user previews the illustration results and saves the outcome

Requirement 10. The system should support the foreground segmentation of a scene**.**

Use Case Title	The system should support figure-ground segmentation.
Brief description	Users can isolate a foreground scene for further editing or emphasis
System input	An image source that depicts an action.
System output	The foreground segmentation of the input image=.
Scenario steps	Step 1. The use selects the visualization tool for the figure-ground segmentation. Step 2. The user loads an edited image source by any of the other tools. Step 3. The user defines the foreground region of interest. Step 4. The user further defines out of context areas of the foreground region of interest. Step 5. The user is able to edit the background in order to amplify the foreground. Step 5. The user previews the results and saves the outcome.

Requirement 11. The system should support the synthesis of the tools' visualizations in the form of comic material.

Use Case Title	Comic material Synthesizer
Brief description	Users can create comic pages using multiple comic grid templates.
System input	Multiple image files
System output	The system exports comic pages including the image inputs.
Scenario steps	Step 1. The user selects a series of edited image files Step 2. The user inserts the images in specified comic grids. Step 3. The user selects to save the comic page in the same project directory.

Requirement 12. The system should support the exportation of the visualization results in multiple output formats.

Use Case Title	Export visualization results in multiple output formats
Brief description	Users can export 2D visualizations in various extension formats (jpg", "jpeg", "png", "tiff", "bmp", "svg", "jfif", "gif")
System input	A list of visualizations and a format for export
System output	The system exports the selected visualizations in the selected format
Scenario steps	Step 1. The user locates and selects the export functionality Step 2. The user selects the sources to be exported from the list of project visualizations Step 3. The user selects the export type from the list of available export types Step 4. The user selects to export the selected visualization in the selected output format

4.3 Design Solutions

This stage focuses on the design of the User Interface (UI) prototypes for the MotiVo editor. It was decided that the editor will be based on a plug-in architecture concept in which multiple standalone tools, in this case, visualization tools, can be embedded and managed under one main system interface. The rationale behind this decision was that it provides ready-to-use components that fit the objectives of the MotiVo editor as well as such an architecture would allow the easy expansion of the system with the incorporation of new visualization tools as they become available in the future. To this end, the UI design of the MotiVo system followed a dockable UI container design approach, which is used by Integrated Development Environments. The dockable UI container approach allows the user to load and unload components on demand and new components can be loaded as plugins by integrating a new dockable window to the main window manager. The dockable UI container paradigm is a common way that common graphics and code editing software making it familiar to users with experience on using such applications. At the same time, the structure of the dockable UI is simple enough even for an intermediate computer user to understand as all tools and assets can be accessed from one screen. Furthermore, this was considered a good option as one of the target groups of these tools is technical people with expertise on using dockable layouts.

Based on the aforementioned design rationale, the design process of the MotiVo editor is separated into two parts. The first part focuses on the design process of the dockable UI of the system, while the second part focuses on the design process followed for each of the implemented and incorporated MotiVo's tools.

Dockable User Interface

For the design of the shell, the free third-party docking library 'DevZest WPF' [96] was used. This library was chosen because it fits perfectly with the design rationale of the MotiVo editor of having one main system, which can incorporate multiple tools and components that support motion artistic visualization processes. Moreover, it was chosen because it provides ready-made UI structural components and useful window management functionality features that are commonly found and used in typical developer-oriented tools (e.g. Visual Studio), such as application window management, docking of windows according to user preferences, floating windows functionality, etc. Using the abovementioned docking library three UI themes were available in the context of the MotiVo editor; light theme (Figure 22), blue theme (Figure 23) and dark theme (Figure 24). The dark theme was chosen as the default theme of the MotiVo editor as the position of the toolbar of the right container is more obvious and easier to access by the user as well as this structure is the most commonly used in other typical graphic software and development applications.

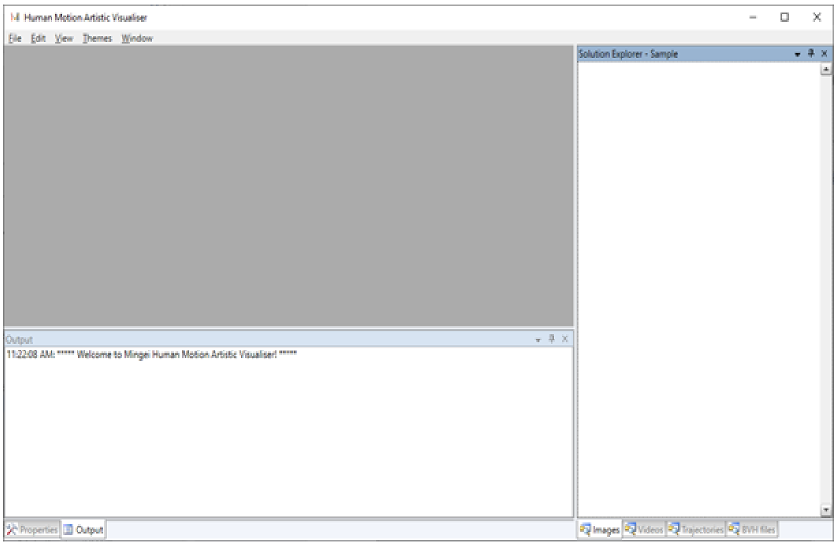


Figure 22. DevZest docking library - Light Theme (VS2008)

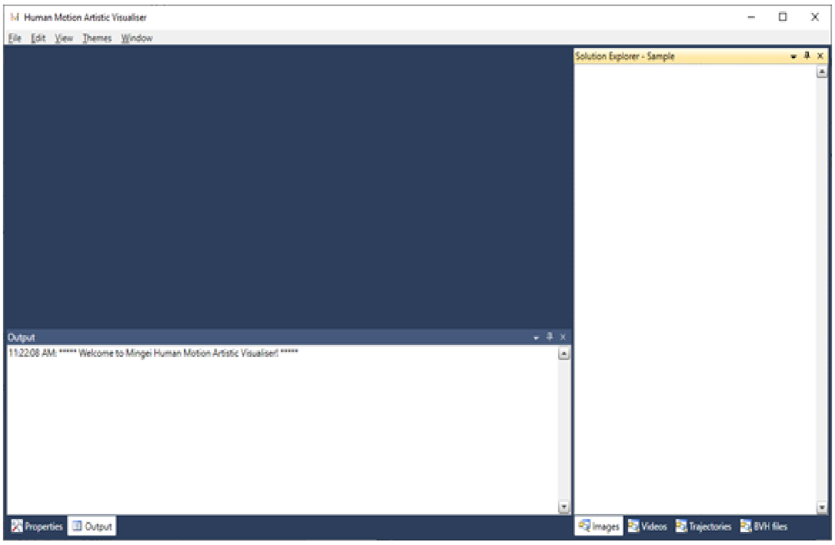


Figure 23. DevZest docking library - Blue Theme (VS2010)

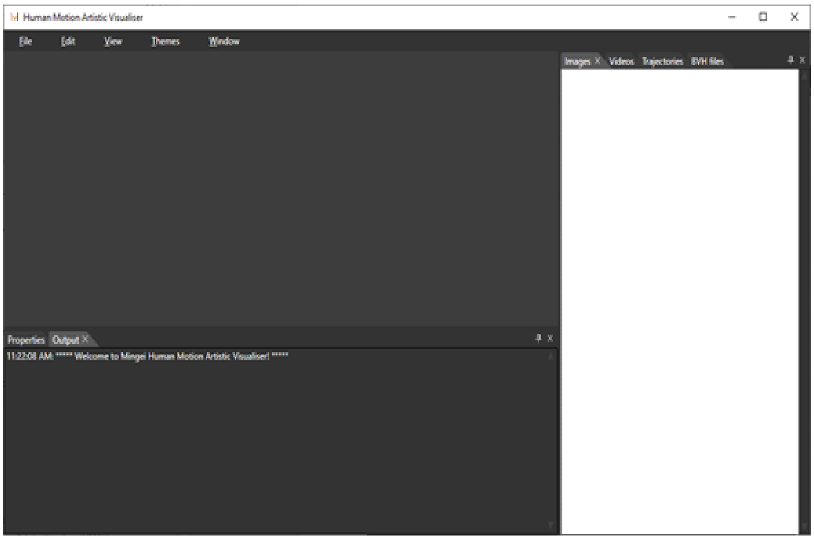


Figure 24. DevZest docking library - Dark Theme

4.4 Evaluate against requirements

Evaluation is an integral part of the UCD methodology and the embedded iterative design approach and it is suggested to take place from the very early stages of the design cycle. Its main aim is to help the design and development team uncover and fix potential usability and user interaction problems introduced in the prototype design at a stage where changes are less costly. During the development of the MotiVo's tools, multiple expert-based evaluations in the form of cognitive walkthroughs were conducted by HCI usability experts, aiming to eliminate serious usability problems before proceeding to user testing. The whole evaluation process is extensively described in Chapter 6.

.

Chapter 5

MotiVo Approach

Inspired by the theory of human perception and the available motion visualization techniques used in visual arts, we developed MotiVo [97], a 2D editor that provides seven (7) visualization tools, each one approaching a different visualization technique, as well as creates static motion visualizations similar to the principles followed in drawings, comics, photography and instruction manuals. The MotiVo editor facilitates workflows between the available tools and generates insightful and artistic results by requiring minimum expertise and knowledge from the user side. The need is to fill the observed gap between motion capture and artistic motion visualization as well as to allow the generation of static motion demonstrations and summarizations, such as books, illustrations, online demonstrations, craft educational material, etc. Specifically, the MotiVo editor consists of the following tools:

1. Motion Blender
2. Foreground Amplifier
3. Image Simplifier
4. Image Filters
5. Trajectory Visualizer
6. Comic Annotator
7. Comic Synthesizer

The MotiVo's tools are extensively presented in Section 5.2.

5.1 Editor Overview

Based on the proposed architecture for a plug-in system-UI architecture, the MotiVo editor was structured on the top of DevZest WPF Docking library [96]. In this architecture, components are loaded and unloaded on the fly and new components can be loaded as plugins by integrating a new dockable window to the main window manager. To that end, the MotiVo's editor tools are loaded as window components and they are drag-drop enabled i.e. they are receiving draggable input in the form of images and 2D trajectory files. DevZest WPF Docking library also simplifies the integration of undo/redo-able tabbed docking, floating and auto-hide window management into WPF applications. Figure 25 presents the high-level architecture of the MotiVo editor. The central component is the docking library shell, in which all the MotiVo's tools are integrated. These tools can create workflows, to that end, the output of each tool can be used as input to any other; in this way, the MotiVo editor can generate hybrid results.

A basic characteristic of the MotiVo system, is that it enables the creation of projects as *.hmav* file types. To that end, users can create libraries with edited material and loaded them any time as assets of the specific project. In order users to create a new project two (2) types of file directories are required; '*Images*' and '*Trajectories*'. Image path is mandatory to be assigned as all the MotiVo's tools take as input image sources whereas trajectory path is optional as there is only one tool that exploits trajectory files.

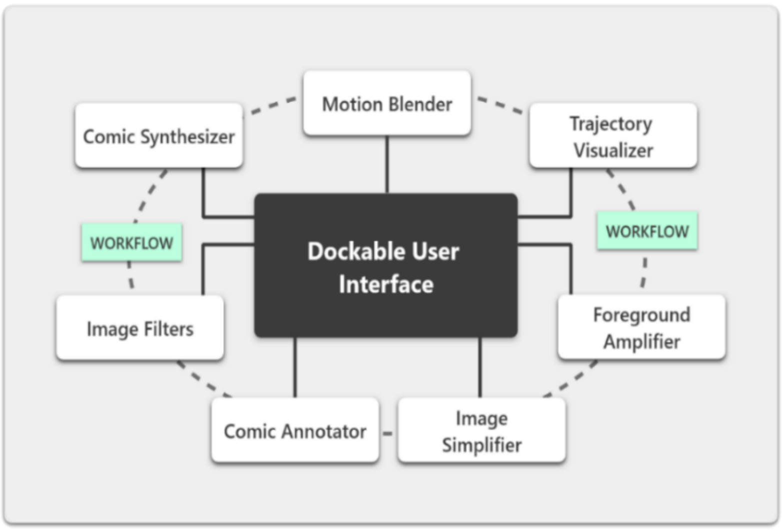


Figure 25. High-Level architecture

Dockable User Interface

The dockable user interface of the MotiVo editor is structured into five (5) distinct sections (Figure 26): (1) Main menu, (2) Tools tab menu, (3) Visualization container, (4) Assets' Library panel (5) Output window and (6) Save and Clear buttons.

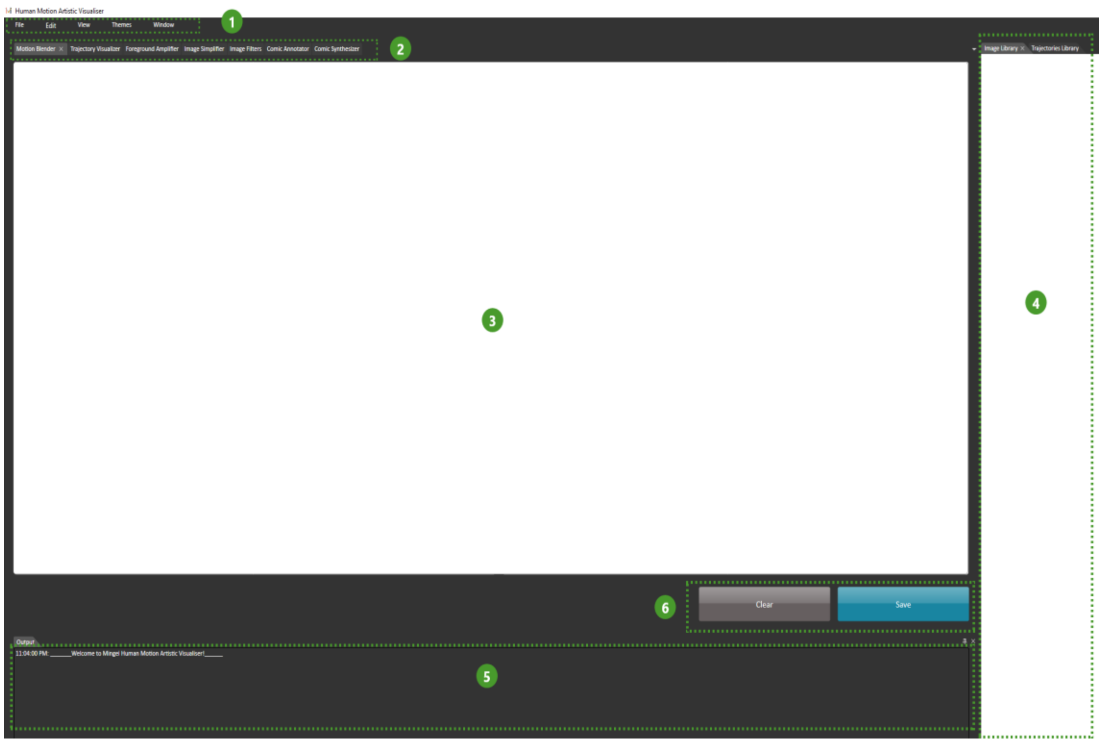


Figure 26. MotiVo UI components

1. Main menu

The main menu provides the file management options as well as secondary options such as themes and edits (Figure 27). Specifically, in the 'File' menu option are located the project functionalities such as creating a new project, opening of an existing project as well as closing the currently loaded project (Figure 28).

Figure 27. Dockable User Interface – Main menu

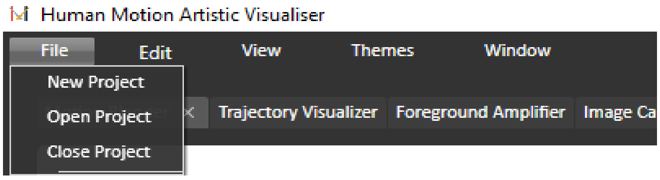


Figure 28. Dockable User Interface – Main menu, 'File' option

For the creation of a new project, a modal window appears that allows users to assign the project name, image directory path and trajectory path (Figure 29). When this information is provided, a file system browser dialogue appears that allows users to select the file name and the location in the file system that the new project will be saved as a 'hmav' file type. In the case of opening an existing project, a typical browsing window appears and users can select a '*hmav'* file type in order to start the editing (Figure 30). The 'Close Project' option, closes the currently loaded project by clearing all the image and trajectory assets from the library as well as the subject matter of each visualization tool.

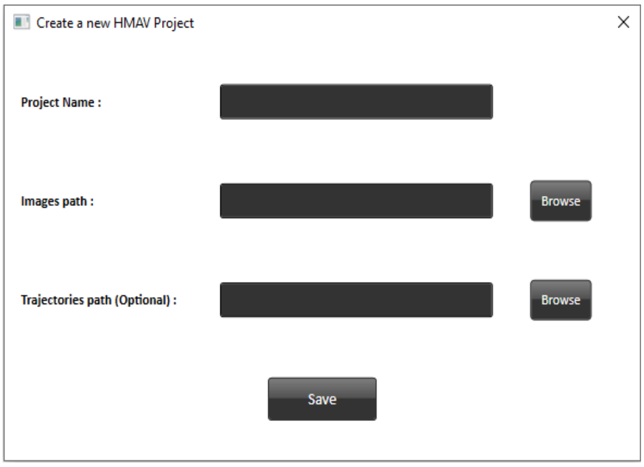


Figure 29. 'New project' functionality – Modal window

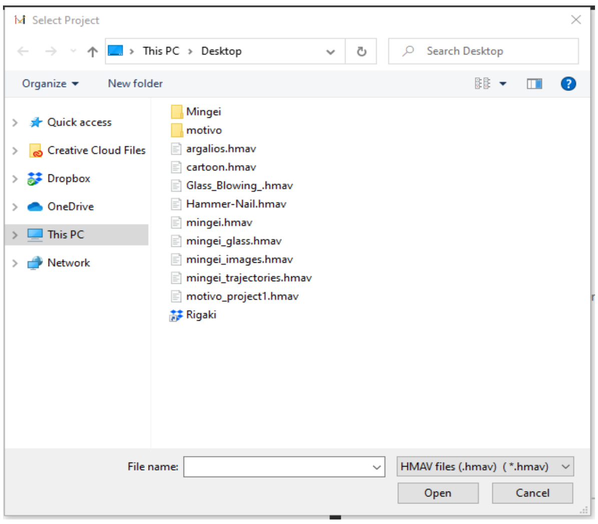


Figure 30. 'Open Project' functionality

2. Tools tab menu

In this menu all the MotiVo tools are located next to each other in the form of tabs (Figure 31).

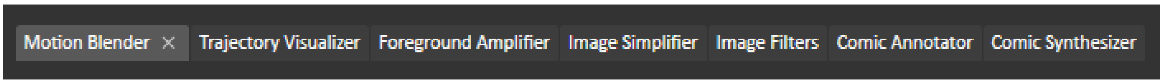


Figure 31. Dockable User Interface – Tool tab menu

3. Visualization container

In the visualization container, the layout of each tool is displayed as well as the visualization results.

4. Assets' Library panel

In this section, the *'Image Library'* and the *'Trajectory Library'* are located in the form of tabs, in which all the project's image and trajectory assets are loaded (Figure 32). These assets are drag-able, thus users can drag and drop any asset on the corresponding tool's visualization container.

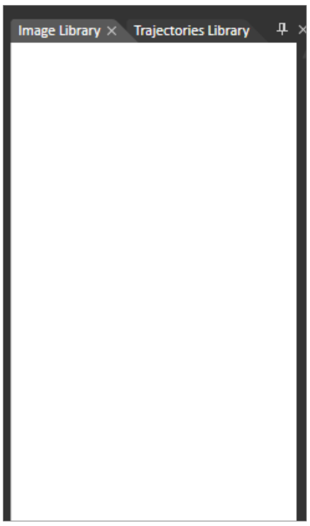


Figure 32. Dockable User Interface - Assets' Library panel

5. Output window

In the 'Output' window, visualization information for each tool is posted (e.g., selected file name, time and tools' name) (Figure 33).

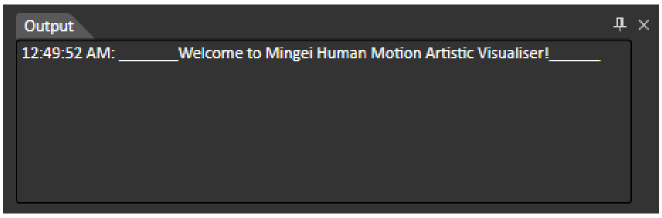


Figure 33. Dockable User Interface – ‘Output’ window

6. Save and Clear buttons

On top of the basic dockable user interface we designed ‘Clear’ and ‘Save’ buttons which are visible for each visualization tool of the MotiVo editor (Figure 34). By clicking the ‘Save’ button the visualization output from every tool is saved in the ‘Image Library’. The ‘Clear’ button wipes out the visualization container (e.g. dropped images, elements).

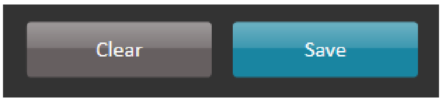


Figure 34. Dockable User Interface - 'Save' and 'Clear' buttons

5.2 MotiVo Tools

In the subsequent sections, the seven (7) tools of the MotiVo editor are extensively described.

5.2.1 Motion Blender

The 'Motion blender' tool was inspired by the visual motion technique of multiple images used in photography as well as superimposition. Specifically, it creates motion summaries on a single image by blending different key poses of a depicted action into unified image content.

- **Input**

  As input, the Motion Blender tool takes a list of key pose images extracted from a video record of a craft or motion activity. Each key pose can be any file extension type of image (i.e. jpg, jpeg, png, gif, tiff, BMP, SVG, JFIF) of the same size.
- **Output**

  As output, the Motion Blender tool produces a single image depicting the summarization of a movement activity.

Regarding the number of input images, their visualization order as well as the intensity weight of each key frame, users can modify them and view the result on runtime. The resultant visualized image, except for the overall motion summarization, designates the direction of any moving process.

This tool supports two (2) types of motion summaries; '*averaged'* and '*weighted'* visualizations, specified by the distribution of the intensity weights on each key pose. The available weight value range for each image is ranged between 0-100, with zero (0) to be the lowest intensity weight and one hundred (100) the maximum intensity. For averaged motion visualizations, user should distribute evenly the intensity values for each image, in contrast to weighted motion visualizations, where the intensity values should be unevenly set. Images set with low intensity weight values are visualized as "faded" in the final result, whereas images with high intensity weight values are more distinct in the final result.

According to the nature of the activity, the emphasis on a specific key pose can be differentiated to accomplish alternative motion visualization results. In some actions, the most significant key pose is either the initial or the last one. For instance, in the process of hammering a nail, the primary key pose could be the last one as it indicates how exactly the hammer should hit the top of a nail (weighted motion visualization). In contrast to the previous scenario, in dance choreographies the overall motion sequence could be considered as important, thus the intensity weight of each key frame should be uniformly set (averaged motion visualization).

In 'Motion Blender', the resulting image summary can be viewed at runtime. By default the extension of the output image is a *.png* file, however, users can change this file extension to any of the aforementioned types.

Notation

A color image *I* at coordinates *x, y* has pixel value *c = I(x, y)* where c has values {R, G, B}. In Motion Blender, we do not treat monochromatic (grayscale) differently. If such input is given, the monochromatic channel is replicated in all the RGB (Red Green Blue) bands and the image is treated as a coloured image.

Averaged Motion Visualization

Let *n* be the number of key frames selected by the user. We denote by *Ii* the corresponding images, where *i* is in [1, n]. In case of even distribution between contrast and volume values of the image frames, for each *Ii,* we average the corresponding pixel RBG values from the colour arrays of each selected image (1). The combined result is a visualization of all the images demonstrating a motion sequence (Equation 1).

$$I_s = \sum_{i=1}^{n} \frac{I_i(x,y)}{n}$$

Equation 1. Averaged motion visualization

Weighted Motion Visualization

An extended approach of the previous case is the weighted motion visualization. In this case, the user-specified values are not evenly set. Let *wi* be the contrast weight for each image. For each RGB pixel value of an image source, we multiply them with the contrast weight value set for each image divided by the total number of weight values to normalize the result. The image formation equation of the resultant image is the following (Equation 2):

$$I_s = \frac{\sum_{i=1}^{n} w_i * I_i(x,y)}{\sum_{i=1}^{n} w_i}$$

Equation 2. Weighted motion visualization

5.2.1.1 User Interface Key Components

The main components that constitute the user interface of the 'Motion Blender' tool are: (1) Image Library, (2) Images drop area, (3) Visualization output (Figure 35).

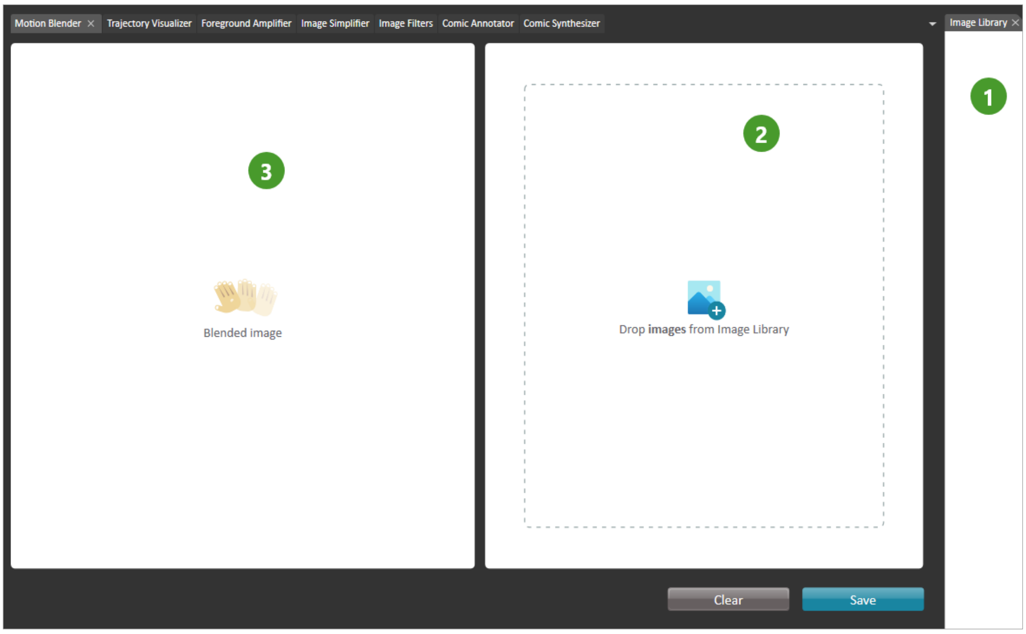


Figure 35. Motion Blender tool - User interface

In the 'Image Library' (i.e. section 1) the images required for the motion blending are listed. In the 'Drop list area' (i.e., section 2) are listed all the dropped images along with a UI slider for each dropped image. The visualization result is displayed at runtime in the motion blending panel (i.e., section 3).

5.2.1.2 Use case

Averaged Motion Visualization

The averaged motion visualization is performed by selecting the 'Motion Blender' tab from the tool bar and then drag-n-drop any image files from the 'Image Library' to the 'Drop list area'. By default, as soon as the user drops the images to the 'Drop area', the intensity weight value for all the images, is automatically set to maximum (i.e., 100). The averaged motion visualization is successfully performed when all the weight values are equal for every dropped image (Figure 36).

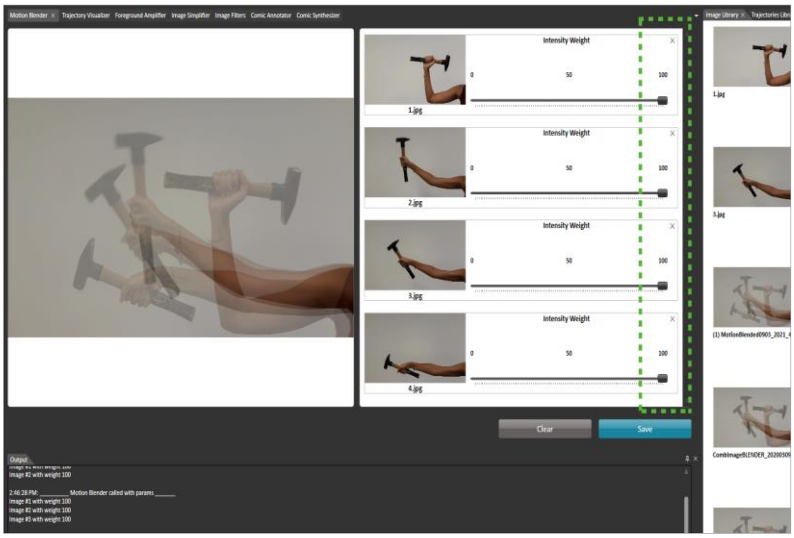


Figure 36.Motion Blender tool – Averaged motion visualization

Weighted Motion Visualization

The weighted motion visualization is performed by altering the intensity weight for each of the selected images. In Figure 37, the emphasis is given to the first key pose of the action as the weight value of the first image is the highest of all. On the contrary, in Figure 38, the last key pose is visualized the most as the weight values for each image are set in ascending order.

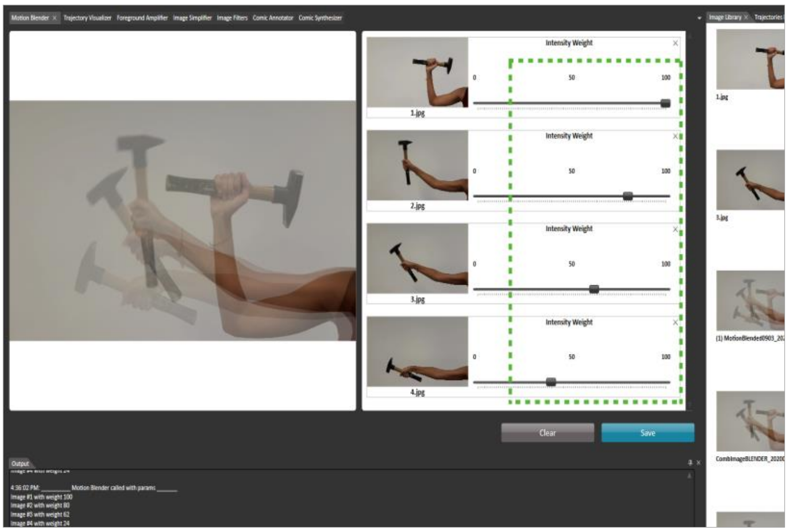


Figure 37.Motion Blender – Weighted Motion Visualization with emphasis on the first key pose.

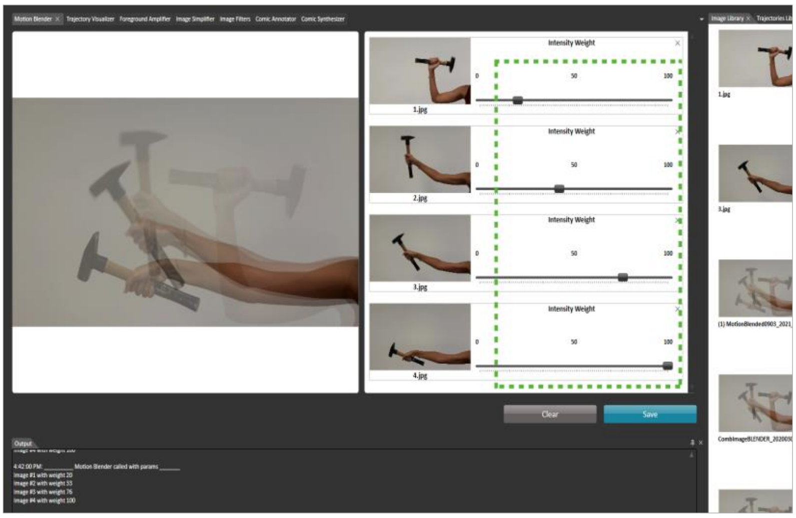
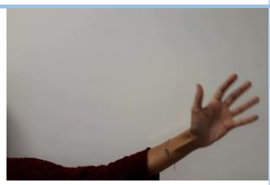


Figure 38. Motion Blender – Weighted Motion Visualization with emphasis on the last key pose.

Examples

We examined three cases in which the contrast weight value of each key pose is differentiated according to the scene and the action.

Weighted Motion Visualization - Emphasis on the initial key pose

Input image #1	Input image #2	Input image #3	Output
			
			
			

Weighted Motion Visualization - Emphasis on the last key pose

Input image #1	Input image #2	Input image #3	Output
			
			
			

Averaged Motion Visualization

Input image #1	Input image #2	Input image #3	Output
			
			
			

5.2.2 Foreground Amplifier

Inspired by the figure-ground segmentation theory, the 'Foreground Amplifier' tool creates foreground/background masks and extracts the foreground region of interest (ROI) from the background of an image. This allows the distinction and amplification of the foreground scene of a movement activity or craft practice, in static mediums.

- **Input**

  As input, the Foreground Amplifier tool takes an image depicting a possible foreground action to be amplified.
- **Output**

  As output, the Foreground Amplifier tool produces a segmented image with amplified the selected foreground scene.

A foreground object refers to any object of interest in an image while the background of an image refers to all pixels that are not part of the foreground object. We focused on foreground extraction in order to accentuate and isolate a region of interest in a scene and produce even better and unique visualization results.

This tool has integrated GrabCut [98], a ready to use image editing algorithm for the foreground and background segmentation provided by the OpenCV library [99]. CrabCut is an innovative segmentation technique that was designed by C. Rother, V. Kolmogorov and A. Blake from Microsoft Research Cambridge and it is based both on texture and edge information as seen on Interactive Graph-cuts [100] but it has further extended this approach by enhancing the graph-cut mechanisms for an improved quality of user interaction and results. Specifically, the GrabCut algorithm takes as an input an 8-bit 3-channel image and produces the segmented form of the input image, defined by the user.

Basic Functionality

As in classic algorithm, in the Foreground Amplifier tool, the user draws a rectangle around the foreground region (foreground region should be completely inside the rectangle) and everything outside this rectangle is considered as sure background. The image region inside the rectangle is still considered unknown. According to the algorithm, the first step is the labelling of the pixels to foreground and background (1 for foreground and 0 for background). Then a Gaussian Mixture Model (GMM) is used to model the foreground and background and creating a new pixel distribution based on the unknown labelled that are either probable foreground (labelled as 3) or probable background (labelled as 2) depending on its relation with the other hard-labelled pixels.

As a result, a weighted graph is built from this pixel distribution and the nodes of this graph are pixels. Source node and Sink node are also added to this graph with every foreground pixel connected to the Source node and every background pixel connected to the Sink node. In the produced graph every edge that connects the pixels to source and end node are defined by weights that characterize the probability of a pixel being foreground /background. Then a Min-Cut/Max-Flow algorithm is used to segment the graph. After the cut, all the pixels connected to the Source node become foreground and those connected to the Sink node become background. In the GrabCut algorithm, a matting technique is also used to calculate the alpha matte for boundaries of segmented regions. Figure 39 illustrates the process of labelling an image, creating the graph, and then segmenting the graph to produce a segmentation of the image.

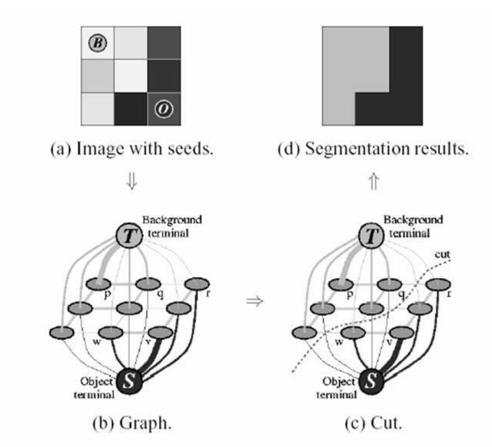


Figure 39.GrabCut- (a) Labelling an image (b) Graph creation (c) Segmentation of the graph (d) Segmented images

In Table 2, the GrabCut function is displayed as it is called in C# as well as the parameters that it takes.

Table 2. GrabCut function parameters provided by OpenCv library

CvInvoke.GrabCut (image, mask, rect, bgdModel, fgdModel, iterCount , mode**)**	
Parameters	
img	input 8-bit 3-channel image
mask	a mask image where we specify which areas are background, foreground or probable background/foreground etc. It is done by the following flags, cv.GC_BGD, cv.GC_FGD, cv.GC_PR_BGD, cv.GC_PR_FGD, or simply pass 0,1,2,3 to image

rect	a rectangle which includes the foreground object in the format (x,y,w,h). The pixels outside of the rectangle are marked as "obvious background". The parameter is only used when mode==GC_INIT_WITH_RECT
bgdModel	temporary array for the background model
fgdModel	temporary arrays for the foreground model
iterCount	number of iterations the algorithm should make before returning the result
mode	Operation mode that could be one of the **GrabCutModes**

◆ GrabCutModes

enum **cv::GrabCutModes**

`#include <opencv2/imgproc.hpp>`

GrabCut algorithm flags.

Enumerator	
GC_INIT_WITH_RECT Python: cv.GC_INIT_WITH_RECT	The function initializes the state and the mask using the provided rectangle. After that it runs iterCount iterations of the algorithm.
GC_INIT_WITH_MASK Python: cv.GC_INIT_WITH_MASK	The function initializes the state using the provided mask. Note that GC_INIT_WITH_RECT and GC_INIT_WITH_MASK can be combined. Then, all the pixels outside of the ROI are automatically initialized with GC_BGD .
GC_EVAL Python: cv.GC_EVAL	The value means that the algorithm should just resume.
GC_EVAL_FREEZE_MODEL Python: cv.GC_EVAL_FREEZE_MODEL	The value means that the algorithm should just run the grabCut algorithm (a single iteration) with the fixed model

In some cases, the segmentation is not accurate as it may have marked some foreground region as background and vice versa. To that end, user interaction is needed to further fine touch-ups by giving some strokes on the images where some faulty results exist. These strokes mark sure foreground and background pixels and thus correct the resultant image in the next iteration (mask parameter).

In the Foreground Amplifier tool, user interaction was our top priority for the foreground extraction as well as the simplicity in the whole process. We have both incorporated the user-based rectangle creation and strokes refinement, all of them provided in a customized toolbar. To that end, for GrabCut Modes, we used both cv.GC_INIT_WITH_RECT and cv.GC_INIT_WITH_MASK.

Another novelty that was added in this tool, is that users can also save the created masks as assets of the loaded project, and thus use the specific masks in the future in order to skip the whole procedure of using *rectangle-cuts* and *mask creations* (i.e. using the foreground and background brushes). In this way, the whole process of segmentation is optimized and the users can easily amplify a foreground scene.

Background Options

Besides those two (2) basic functionalities (i.e., rectangle cut, mask cut), in Foreground Amplifier we have developed a series of extra options for the end-users. Specifically, background options are available giving extra emphasis on the foreground. As soon as, the foreground has been successfully isolated, the background editing options are; blurred, grey blurred, white, black, grey and transparent background.

All the background options have been developed in such a way that the generated mask before the background editing, is smoothed by using the 'Blur' function offered by the OpenCV library in order for the final foreground scene to adapt naturally in the final result. The novelty behind this functionality is that the system has already saved the cut image and by checking the pixel colour and intensity (background is black and foreground is coloured) it can quickly change the pixel colors and create artistic results. Users can instantly view the result without the arduous user interaction that many image software editors may require.

Eraser Option

The Foreground Amplifier tool also provides the option of an eraser which can be used in cases that background fragments still remain in the background of the resultant segmented image. The eraser option sets the coloured pixels of the background as sure background (labelled as 0) and thus remove them from the final result.

Temporarily, this functionality is available only for black and white background options. An eraser option was included for users to easily improve the final extracted foreground scene and create more aesthetic results.

5.2.2.1 User Interface Key Components

The main components that constitute the user interface of the 'Foreground Amplifier' tool are: (1) Image Library, (2) Drop image area, (3) Rectangle area, (4) Mask area, (5) Background Options, (6) Eraser area (Figure 40).

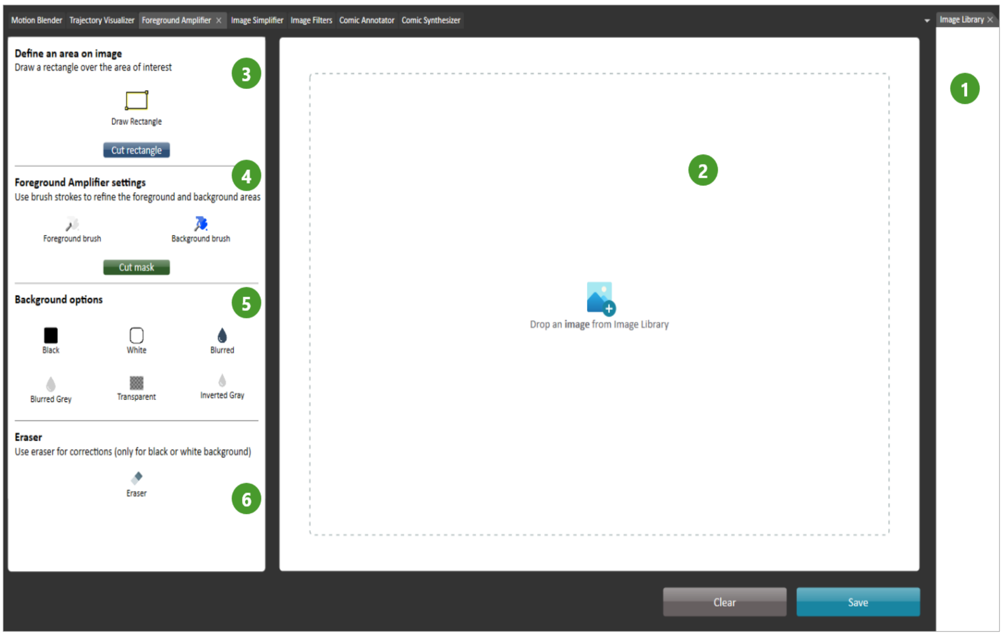


Figure 40. 'Foreground Amplifier' tool User Interface

In the 'Image Library' (i.e. section 1) the images required for the foreground amplification are listed. In the 'Drop image area' (i.e., section 2) is displayed the dropped input image. In the 'Rectangle area' (i.e., section 3) the functions for drawing a rectangle around a foreground object as well as *'cut'* the specific rectangle are located. Specifically, in section 3, there are two (2) buttons, the 'Draw rectangle' button which enables the user to create a rectangle around an area of interest by using the mouse cursor, and the 'Cut rectangle' button which extracts the defined foreground and updates the input image in section 2.

The 'Mask area' (i.e.section 4) provides options for creating a mask via two (2) brush options. In more detail, in this section there are three (3) buttons, the 'Foreground brush' enables the users to draw white strokes over the foreground parts of the input image, the 'Background brush' enables the users to draw blue strokes over the background parts of the input image and finally the 'Cut mask' button that generates the final segmentation.

In the section of the 'Background options' (i.e., section 5), multiple background options are available such as black, white, blurred, blurred grey, transparent and grey.

Finally, in the 'Eraser area' (i.e., section 6) there is an 'eraser' button that users can click in order to remove background fragments from the segmented image using the mouse cursor. The eraser button is enabled only in cases of black and white backgrounds.

5.2.2.2 Use Case

The foreground amplification is performed by selecting the "Foreground Amplifier" tab from the tool bar and then drag-n-drop any image from the 'Image Library' to the 'Drop image area'.

Basic Functionality

The first step is to define the foreground area using the ‘Draw rectangle’ button (Figure 41). After drawing the rectangle, the ‘Cut rectangle’ button is enabled.

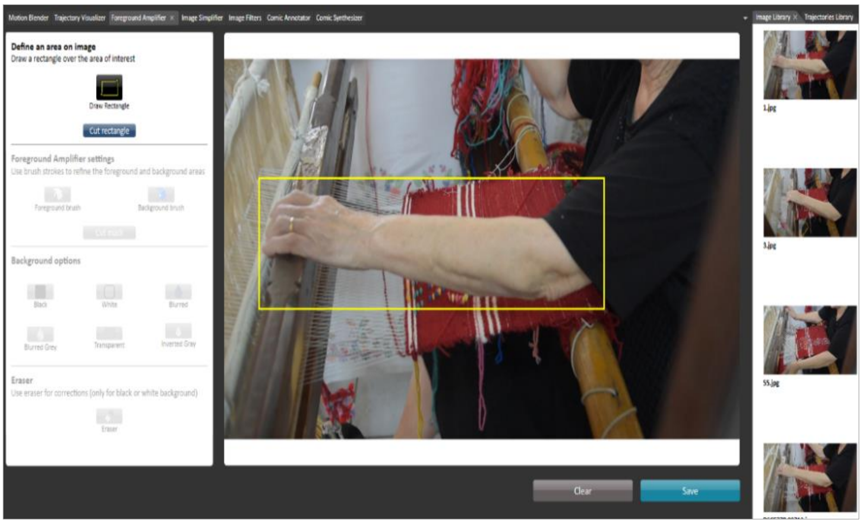


Figure 41. ‘Foreground Amplifier’ tool – Draw Rectangle

Next, the user selects the ‘Cut rectangle’ button and the algorithm is called while the input image is updated (Figure 42). The first basic functionality ‘rectangle cut’ has been completed in this step.

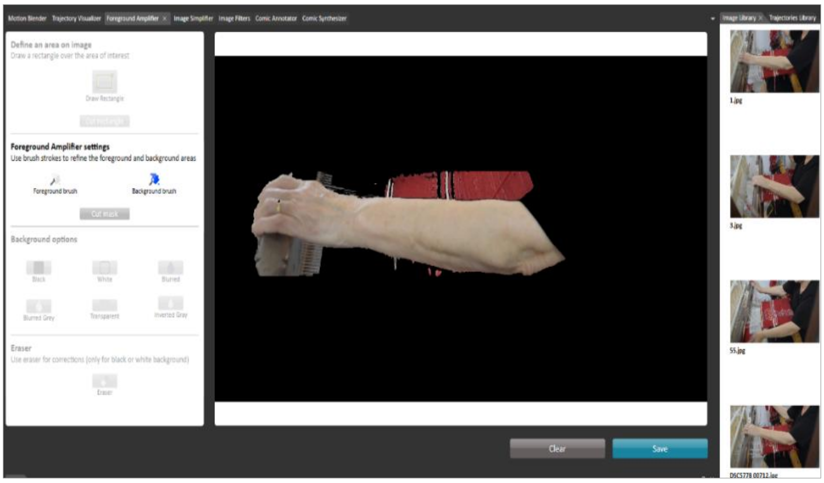


Figure 42. ‘Foreground Amplifier’ tool - Cut rectangle functionality

The third step is the foreground/background refinement on the top of the updated image. White strokes define the foreground scene that the user wants to amplify while blue strokes define the background areas that need to be excluded from the foreground scene (Figure 43).

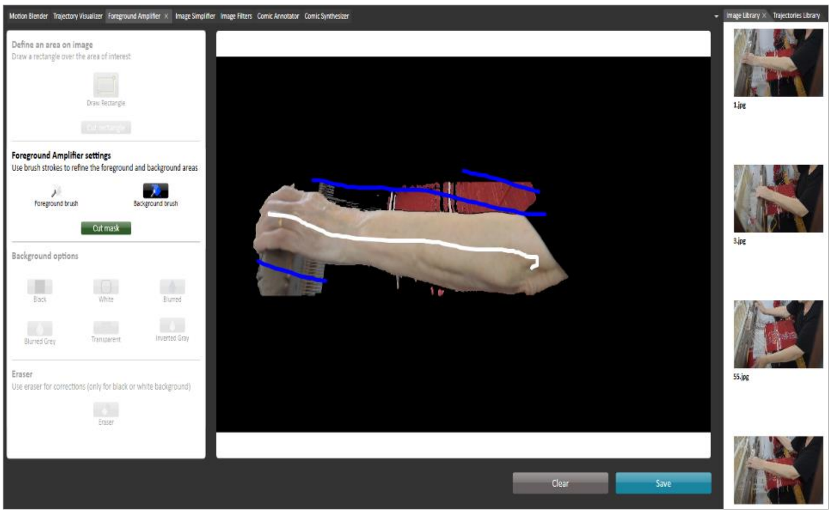


Figure 43. ‘Foreground Amplifier’ tool - Foreground and Background refinement functionality

Once the user has used the strokes to refine the foreground/background areas, now the ‘Cut mask’ button is enabled and by clicking it the algorithm generates the new image segmentation (Figure 44). The basic functionality ‘mask cut’ has been completed in this step.

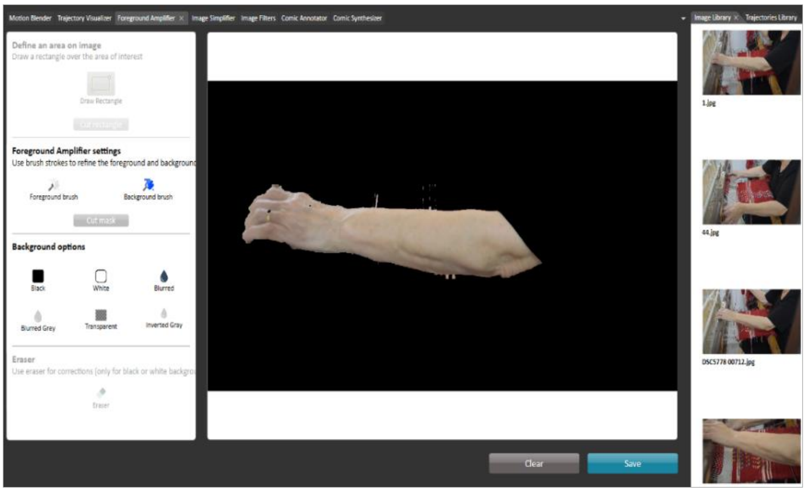


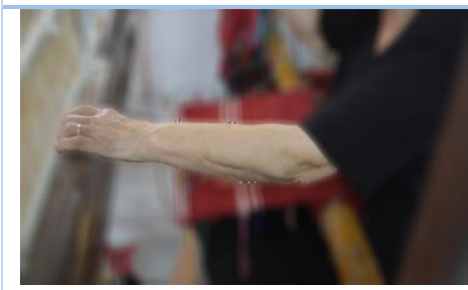
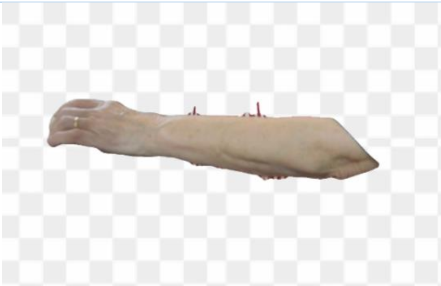
Figure 44. ‘Foreground Amplifier’ tool - Cut mask functionality

Background Options

After the ‘cut mask’ functionality has been completed, the background options are enabled.

Table 3 presents the background options for the example image.

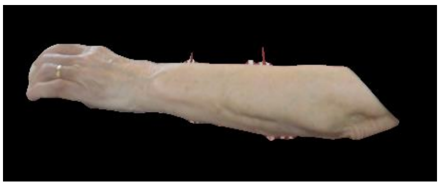
Table 3. Foreground Amplifier – Background Options

Eraser

The eraser option is enabled in cases of black and white background. In Table 4 it is presented the result of using the eraser option.

Table 4. Foreground Amplifier – Eraser Option

Before Eraser	**After Eraser**
	

5.2.3 Image Simplifier

Inspired by the technique of texture used in paintings as well as by comics and cartoons, the 'Image Simplifier' tool has been developed for the creation of simplified image contents that can be easily used for artistic and comic-like material.

- **Input**

  As input, the Image Simplifier tool takes an image.
- **Output**

  As output, the Image Simplifier tool produces a simplified version of the original input image.

The reason behind the integration of this tool in MotiVo editor, is to produce simplified (i.e. comic-like) visualizations that convey excitingly, a craft process or technique. For the development of this tool, we integrated the Rolling Guidance Filter [101], a scale-aware filter that can remove different levels of details in any natural images under a scale measure. The algorithm behind this filter is iterative and can converge quickly and at low cost, fitting perfectly with our vision for the MotiVo editor; ease-of- use and simplicity.

Specifically, Rolling Guidance Filter is a filtering framework that smooths image details according to its scale information while refining edges that can be preserved. In this approach [102] two types of structure scale images are examined as input; small and large scale structure information. Small structures are interwoven with details, noise, small objects and content classified as texture, in contrast to large-scale information that generally encodes boundaries, flat regions and slow spatial colour transition. Initially, Rolling Guidance Filter uses Gaussian filtering to smooth the input image texture but it has been observed [102] that for small-scale structured images, the edges were completed removed according to the Gaussian average mechanism. Nevertheless, large-scale structures were only blurred instead of eliminated. To that end, the first step of the this filter method is the small structure removal that was based both on the Gaussian filter and a constant value that denotes the standard deviation; structures with a scale smaller than this constant are completely removed. The second step of the process is the iterative edge recovery. For the edge recovery, the output of the Gaussian filtering is used. By employing iteratively joint bilateral filter [103] the guidance image changes (Figure 45).

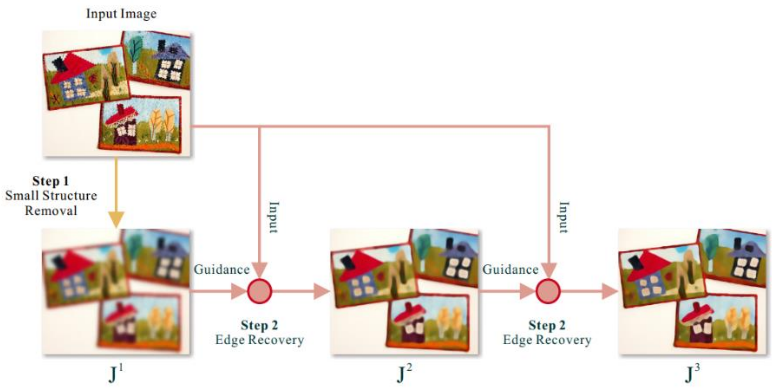


Figure 45. Rolling Guidance Filter - Flow chart for the two steps[102]

In the Image Simplifier tool, this procedure has been designed by adopting user-friendly interface artefacts (i.e. sliders) that enable the effortless creation of valuable image simplified results. The two steps of the Rolling Guidance Filter procedure (Figure 46), in this tool are defined by the user-specified values. The small structure removal step corresponds to the 'Step 1: Structure Removal' which includes one (1) slider labelled as 'Blur' and ranges between zero (0) and ten (10). The edge recovery of adopted to the Image Simplifier tool as 'Step 2: Edge Recovery' and includes two (2) sliders; the first is labelled as 'Simplification' and ranges between zero (0) and thirty (30) and the second is labelled as 'Intensity' and ranges between zero (0) and ten (10) (Section 5.3.4.1).

5.2.3.1 User Interface Key Components

The main components that constitute the user interface of the 'Image Simplifier' tool are: (1) Image Library, (2) Drop image area, (3) 'Structure Removal' area, (4) 'Edge Recovery' area (Figure 46).

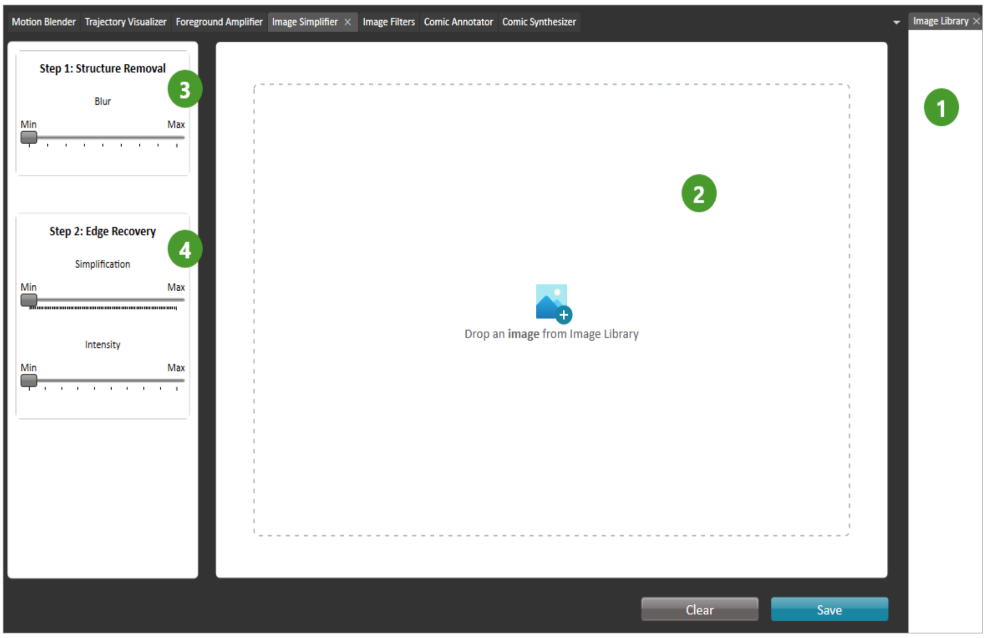


Figure 46. 'Image Simplifier' tool User Interface

In the 'Image Library' (i.e. section 1) the images required for the simplification are listed. In the 'Drop image area' (i.e., section 2) is displayed the dropped image source. The 'Structure Removal area' (i.e., section 3) is the first step of the image simplification process; in this step, the different levels of detail of image source are eliminated (e.g.blurred) via the available user interface slider. The 'Edge Recovery area' (i.e., section 4) is the second step of the image simplification process and it includes two (2) sub-functionalities that are correlated; the 'Simplification' functionality which defines the amount of simplification on the input image and is provided by a user interface slider as well as the 'Intensity' functionality that is also provided by a slider and defines the simplification intensity in the final image.

5.2.3.2 Use Case

The image simplification is performed by selecting the 'Image Simplifier' tab from the tools bar and then drag-n-drop any image from the 'Image Library' to the 'Drop image area'.

Small Structure Removal (Step 1)

As soon as an image is dropped in the 'Drop image area', the Step 1 in the left panel is enabled (Figure 47).

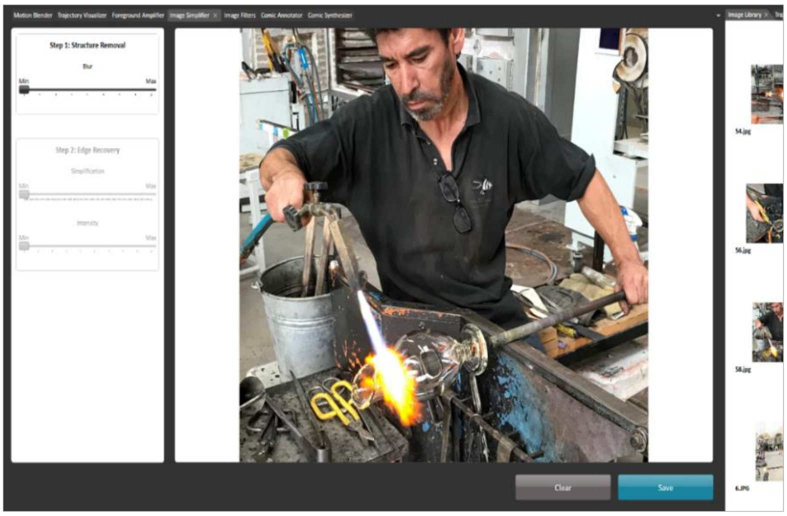


Figure 47. 'Image Simplifier' tool - Input image, before editing

By altering the 'Blur' slider value, step 1 has been successfully completed as the input image content has been blurred (Figure 48). Next, the 'step 2' is enabled.

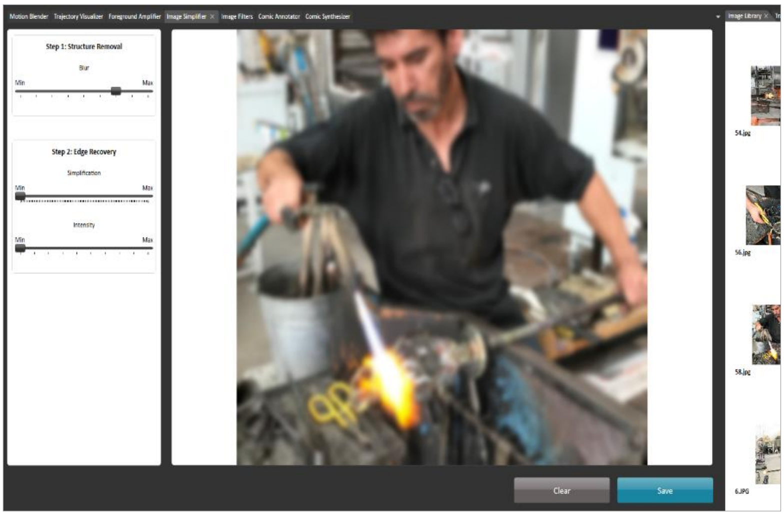


Figure 48. 'Image Simplifier' tool – Step 1: Structure Removal

Edge Recovery (Step 2)

For this step, the 'Intensity' slider value must be defined while the 'Simplification' value is not mandatory to be set. Nevertheless, if only the 'Intensity' value is set, the result would not be distinct, thus it is considered a good practice to set both slider values. As soon as both slider values are set, the simplified image is displayed on the 'Drop image area' (Figure 49).

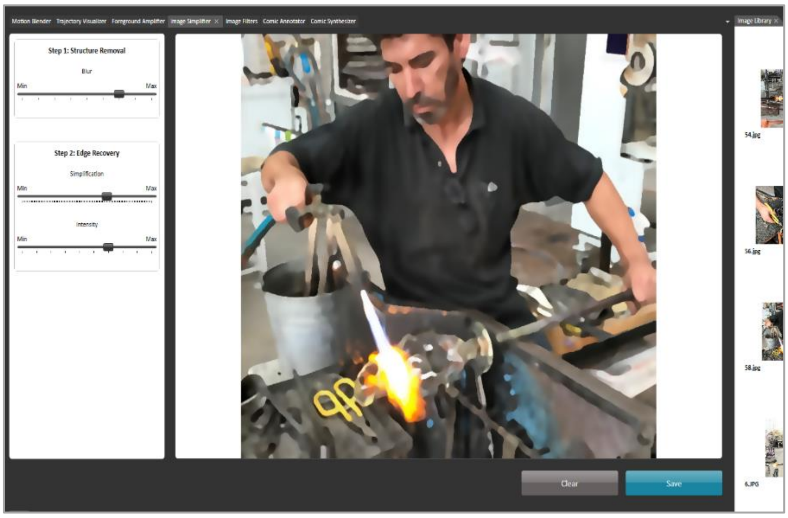


Figure 49. ‘Image Simplifier’ tool- Step 2: Edge Recovery

5.2.4 Image Filters

The artistic visualization as part of the MotiVo editor is integrated by pre and post-processing of images offered by the Image Filters tool. This tool was inspired by the colour contrast and texture techniques used in paintings and supports several specially developed image filters similar to the ones used in popular image processing software adapted to the needs of artistic visualization. The application of these filters alters the appearance of the input image to imply or mimic an artistic concept.

- **Input**

  As input, the Image Filters tool takes an image.
- **Output**

  As output, the Image Filters tool produces a filtered version of the original input image.

This tool combined with other of MotiVo’s editor tools (e.g. Image Simplifier, Motion Blender etc.) produces artistically visualized outputs. For the needs of the MotiVo editor, we integrated a list of numerous image filters such as:

1. Grayscale and Shadows
2. Simple Colour Operations
3. Flip options
4. Image Operators
5. On Canvas
6. Comics
7. Drawing
8. Posters
9. Pop art

These nine (9) categories of image filters were chosen for MotiVo editor as they fitted its objectives and they mainly process the texture and colour contrast of the input images. The complete list of image processing operations is presented in Table 5. This table presents the filter along with their implementation details.

Table 5. 'Image Filters' tool – Filter categories

Grayscale and Shadows	
Grayscale	AForge Image Processing Library
Grayscale to RGB	
Binary Shadows	
Simple Colour Operations	
Invert	AForge Image Processing Library[1]
Rotate	
Sepia	
Extract Red Channel	
Extract Green Channel	
Extract Blue Channel	
Red	
Green	

[1] http://www.aforgenet.com/

Blue	
Cyan	
Magenta	
Yellow	
Flip	
Flip Horizontal	Custom Pixel Shader running on the GPU
Flip Vertical	
Flip Both Directions	
Image Operators	
Blur	AForge Image Processing Library
Sharpen	
Smoothing	
On Canvas	
Canvas	AForge Image Processing Library
Canvas Grayscale	
Canvas Sepia	
Comics	
16 Colours	AForge Image Processing Library
32 Colours	
64 Colours	
Drawing	
Thin lines	AForge Image Processing Library
Medium lines	

Thick lines	
In colour(Thin Lines)	
In colour(Medium Lines)	
In colour(Thick Lines)	
Sketch pencil	
Posters	
Small regions	AForge Image Processing Library
Medium regions	
Large regions	
Pop Art	
Pop Art (Red Green Blue)	AForge Image Processing Library
Pop Art (Pink Violet Orange)	
Pop Art (Yellow Mauve Orange)	
Pop Art (Pink Blue Orange)	

5.2.4.1 User Interface Key Components

The main components that constitute the user interface of the 'Image Filters' tool are: (1) Image Library, (2) Drop image area, (3) List of image filters categories, (4) 'Clear Filter' button (Figure 50).

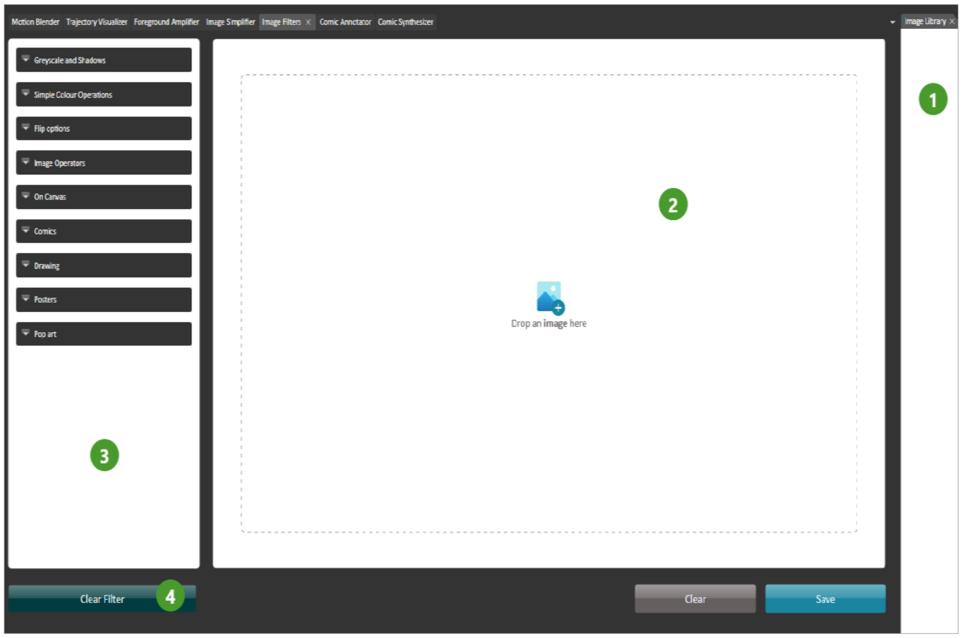


Figure 50. ‘Image Filters’ tool User Interface

In the ‘Image Library’ (i.e. section 1) the images required for the image filtering are listed. In the ‘Drop image area’ (i.e. section 2) is displayed the dropped input image. The ‘List of image filters categories’ (i.e. section 3) provides nine (9) drop-down category menus of different filter stylings. Finally, the ‘Clear Filter’ button (i.e. section 4) ‘clears’ the applied filter on the input image.

5.2.4.2 Use Case

The image filters are applied by selecting the ‘Image Filters’ tab from the tools bar and then drag-n-drop any image from the ‘Image Library’ to the ‘Drop image area’ (Figure 51).

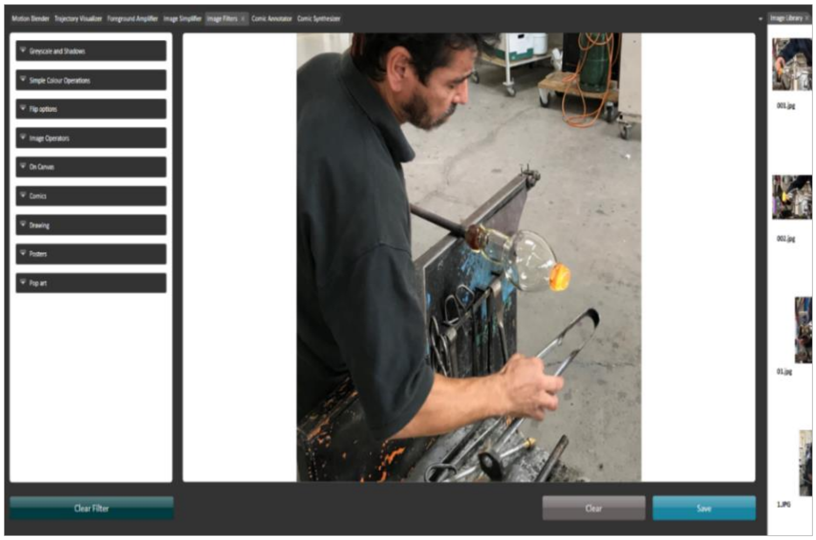


Figure 51.Image Filters – Dropped input image

Each image filter category is a dropdown menu containing the image filter styles. By selecting any of the dropdown image filter options, the source image is automatically updated (Figure 52).

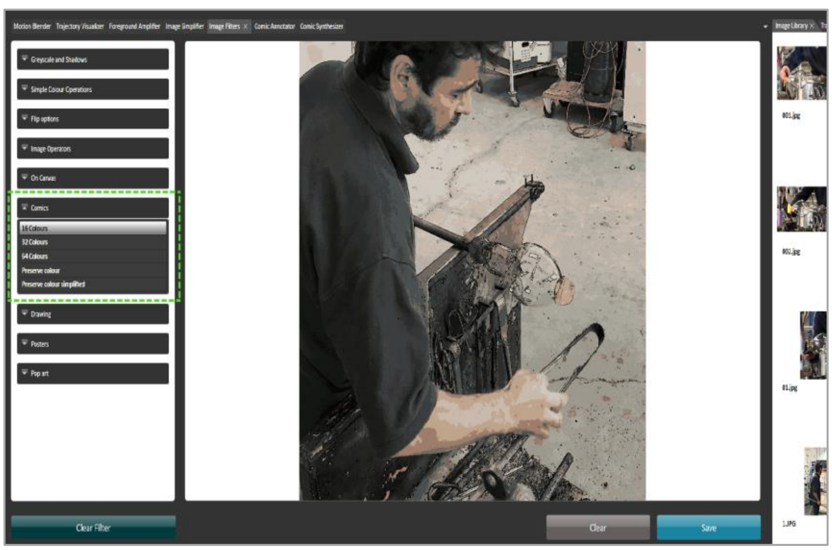
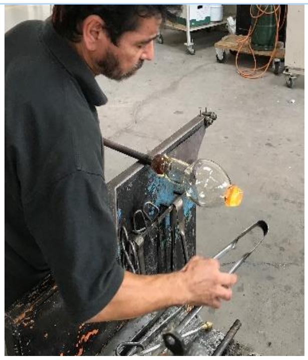


Figure 52. ‘Image Filters’ tool – ‘Comic’ category: ‘16colors’ filter applied

Examples

In this section, we present some of examples regarding the application of several image filters on the same source image (Table 6).

Table 6. 'Image Filters' tool – Filter categories examples

	
Source image	**Posters: 'Medium regions'**
	
Simple colour operations: 'Cyan'	**Simple colour operations: 'Sepia'**
	
Drawing: 'Thick lines'	**Comics: '16 colours'**

	
Pop-art: 'Pink Violet Orange'	**On Canvas: 'Canvas Grayscale'**

5.2.5 Trajectory Visualizer

Inspired by the Gestalt law of continuity, the Trajectory visualizer tool was developed for the visualization of the trajectory joints of an action displayed on a static image. Specifically, it utilizes trajectory files that contain 2D joints of any skeleton body as coordinates of the X and Y axis. These coordinates correspond to the pixels position of the input image. The visualized trajectory denotes the motion direction.

- **Input**

  As input, the Trajectory Visualizer tool takes an image depicting a human activity (e.g. any output image of Motion Blender) as well as, a trajectory file.
- **Output**

  As output, the Trajectory Visualizer tool produces a single image with the trajectory visualization on top of the input image.

The trajectory files have .txt file format and were manually extracted for specific paradigms of this Thesis. The algorithm we have developed, highlights the specific coordinates as well as it uses interpolation and constructs new data points within the range of the discrete set of the known points within the trajectory files. For the purpose of the interpolation, we used composite Bézier curves [104]. In computer graphics, a composite Bézier curve is a piecewise Bézier curve that is at least continuous. The Trajectory Visualizer tool highlights the main trajectory of an action as a red curve and the trajectory points within the trajectory files as black rectangles. All the new constructed points are visualized as blue and green points. This tool is directly related to the Motion Blender tool as it is suggested to take as input the output results of the Motion Blender.

5.2.5.1 User Interface Key Components

The main components that constitute the user interface of the 'Trajectory Visualizer' tool are: (1) Image Library, (2) Trajectory Library, (3) Drop image area, (4) Drop trajectory area (Figure 53).

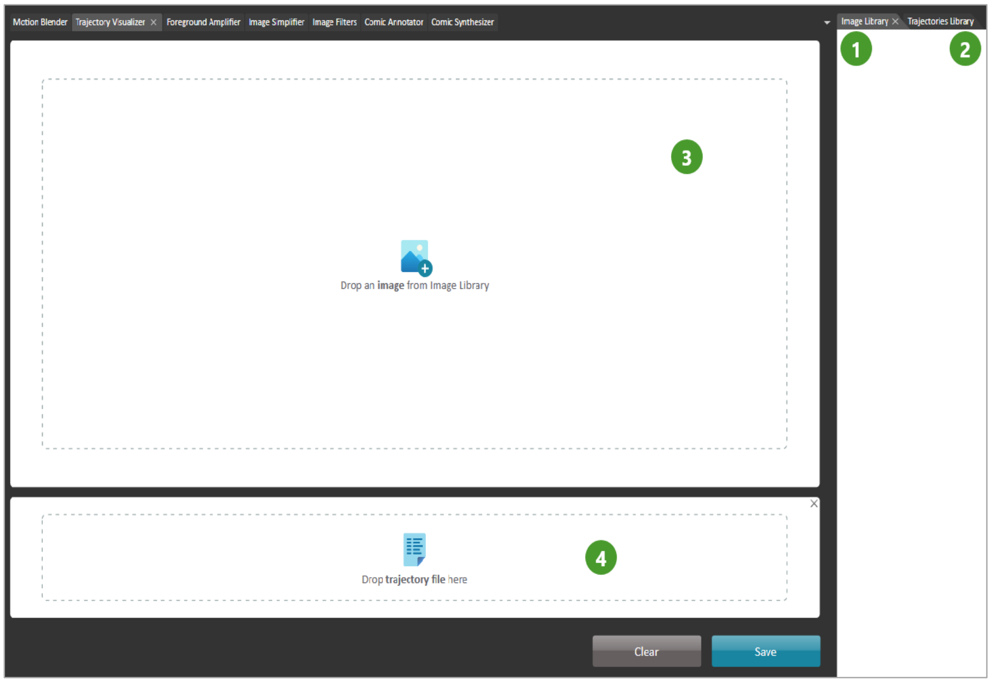


Figure 53. 'Trajectory Visualizer' tool User Interface

In the 'Image Library' (i.e. section 1) the images required for the trajectory visualization are listed while in the 'Trajectory Library' the trajectory files are listed. In the 'Drop image area' (i.e., section 3) is displayed the dropped input image while in the 'Drop trajectory area' (i.e., section 4) the trajectory files from the 'Trajectory Library' are dropped. Once a trajectory file is dropped, it is automatically highlighted on top of the input image.

5.2.5.2 Use case

The motion trajectory visualization is performed by selecting the "Trajectory Visualizer" tab from the tool bar and then drag-n-drop any image file from the 'Image Library' to the 'Drop image area'. The next step is to drag and drop a trajectory file from the 'Trajectory Library' to the 'Drop trajectory area'. As soon as both actions are completed (image and trajectory drops), the Trajectory Visualizer tool dynamically highlights the trajectory's coordinates on top of the input image (Figure 54).

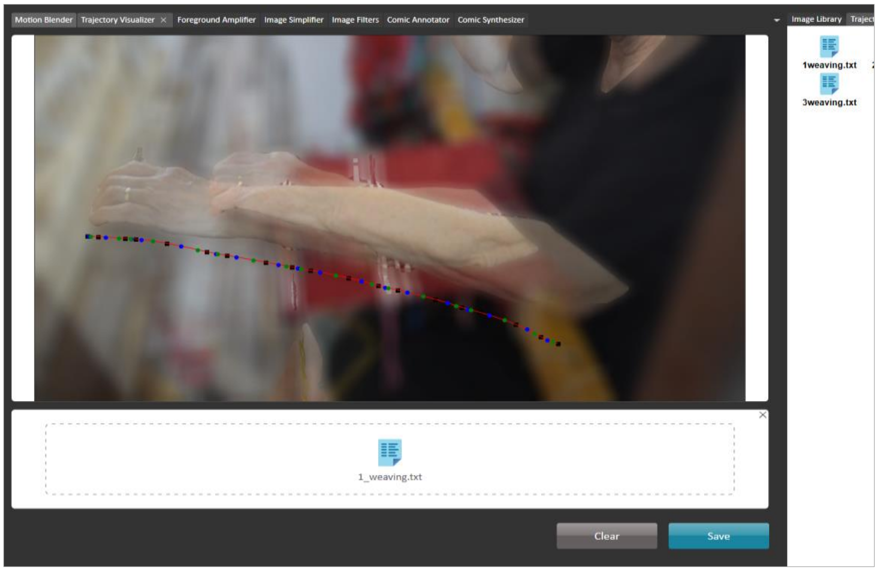
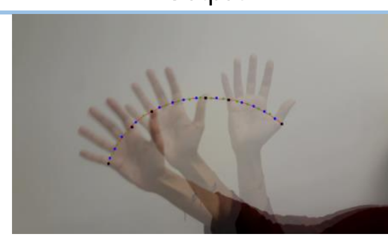
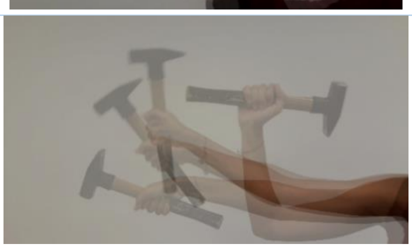
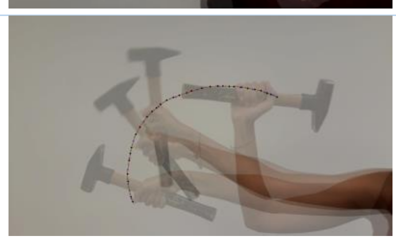
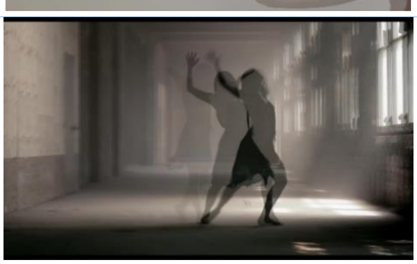
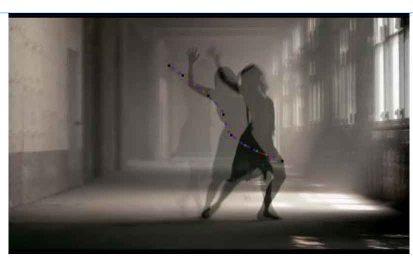


Figure 54. Trajectory Visualizer – Argalios craft

Examples

In this section, we present some examples regarding the visualization of trajectories on different source images (Table 7).

Table 7. 'Trajectory Visualizer' tool – Examples

Input image	Output
	
	
	

5.2.6 Comic Annotator

To exploit the techniques used for superimposition in comics and instruction manuals, we developed the 'Comic Annotator' tool. This tool supports the need for element annotations, especially where semantic information is to be added to the visualization. In the Comic Annotator tool, users manually integrate semantically rich information on static motion visuals by using ready to use concepts such as icons sets and visual elements (e.g. arrows, speech bubbles, text).

- **Input**

  As input, the Comic Annotator tool takes an image file.
- **Output**

  As output, the Comic Annotator tool produces an annotated image.

The idea behind Comic Annotator was based on the law of common fate for 2D static motion visualizations as it can simulate motion when no actual motion exists in a static frame. Specifically, this tool provides three icon library sets:

- Hand Drawn Arrows
- Comic Arrows
- Comic Elements

'Hand Drawn Arrows' were included for artistic and minimal image annotations whereas 'Comic Arrows/Elements' are addressing the need for the creation of comic and cartoon material. Either of the icons can be manually rotated, deleted or even adjust their projection to indicate the depth and direction of a movement activity or pattern.

Apart from the icon elements that can be annotated on top of the source image, the Comic Annotator also facilitates the insertion of both comic and bubble texts. For comic text, by default, is created a textbox with pre-set colours and font style similar to those in comic text boxes. Bubble text, is a simple textbox that can be annotated on a bubble icon when there is a need for dialogues. The Comic Annotator tool can be in cases of instruction manuals and comic pages etc.

5.2.6.1 User Interface Key Components

The main components that constitute the user interface of the 'Comic Annotator' tool are: (1) Image Library, (2) Drop image area, (3) List of icon sets, (4) 'Comic text button and (5) 'Bubble text' button (Figure 55).

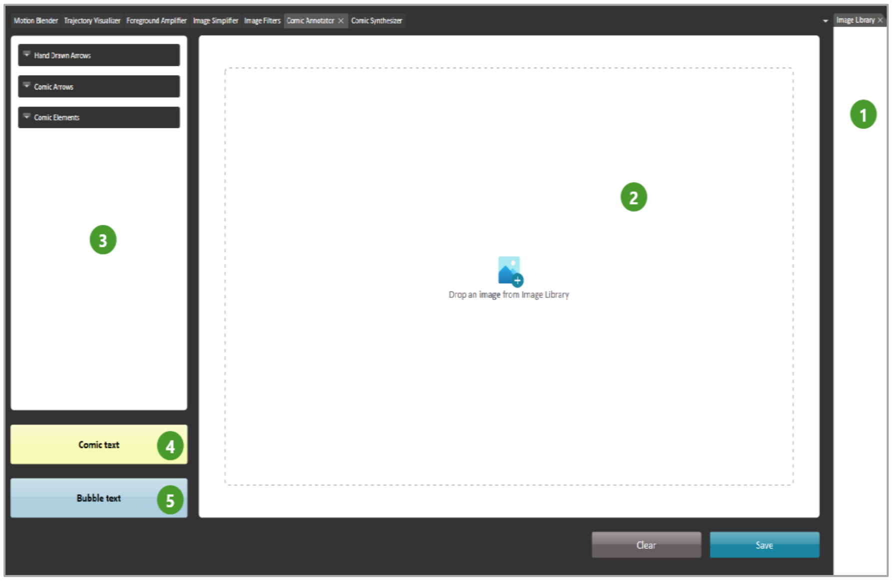
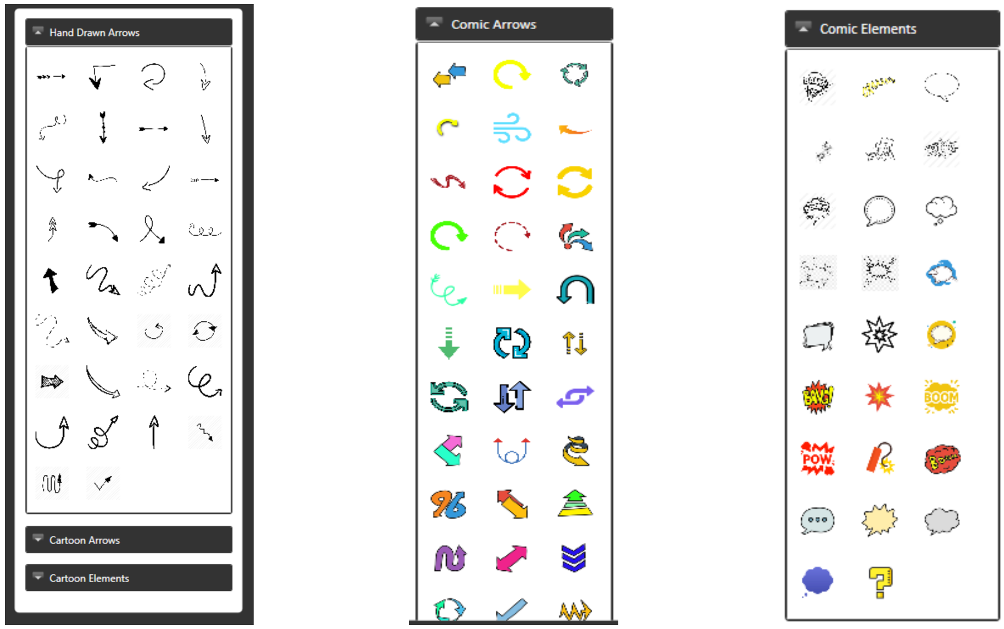


Figure 55. ‘Comic Annotator’ tool User Interface

The ‘Image Library’ (i.e. section 1) contains the images required for the image filtering. In the ‘Drop image area’ (i.e. section 2) is displayed the dropped input image. The ‘List of icon sets’ (i.e. section 3) provides nine (3) drop-down icon menus of different style elements. The ‘Comic text’ button (i.e. section 4) inserts a comic textbox with pre-set colours and font style on the bottom of the input image. Finally, the ‘Bubble text’ (i.e. section 5) button creates a plain textbox on top of the input image that users can annotate inside a bubble icon. In Figure 56, the three (3) available icon sets are presented. Each of these elements are drag-able, thus the user can drop them on the top of the input image and modify them by right-click.



(a) (b) (c)

Figure 56. ‘Comic Annotator’ tool - Icon sets: (a) Hand drawn Arrows (b) Comic Arrows (c) Comic Elements

5.2.6.2 Use Case

The comic annotation is performed by selecting the ‘Comic Annotator’ tab from the tools bar and then drag-n-drop any image from the ‘Image Library’ to the ‘Drop image area’ (Figure 57). Next, users can select any of the available options from the visualization icon sets such as arrows, dialogues or motion lines and drop them on the top of the image source (Figure 58).

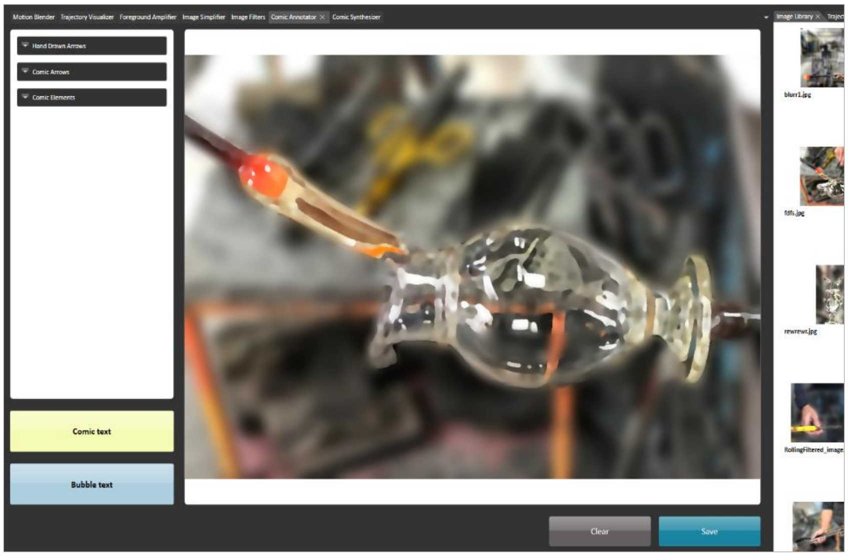


Figure 57. ‘Comic Annotator’ tool – Dropped image source

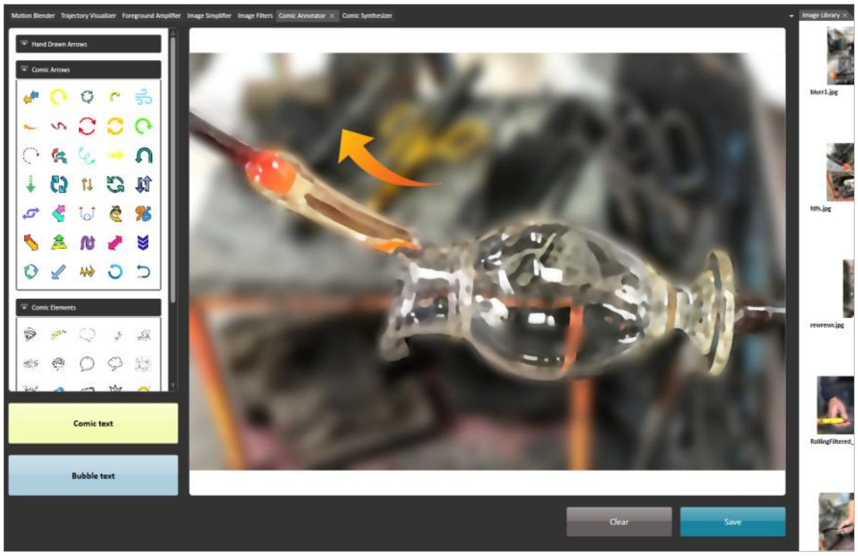


Figure 58. ‘Comic Annotator’ tool – Icon annotation

When the user clicks on the ‘Comic text’ button a yellow textbox is inserted on the bottom of the source image and the user can type anything inside it (Figure 59).

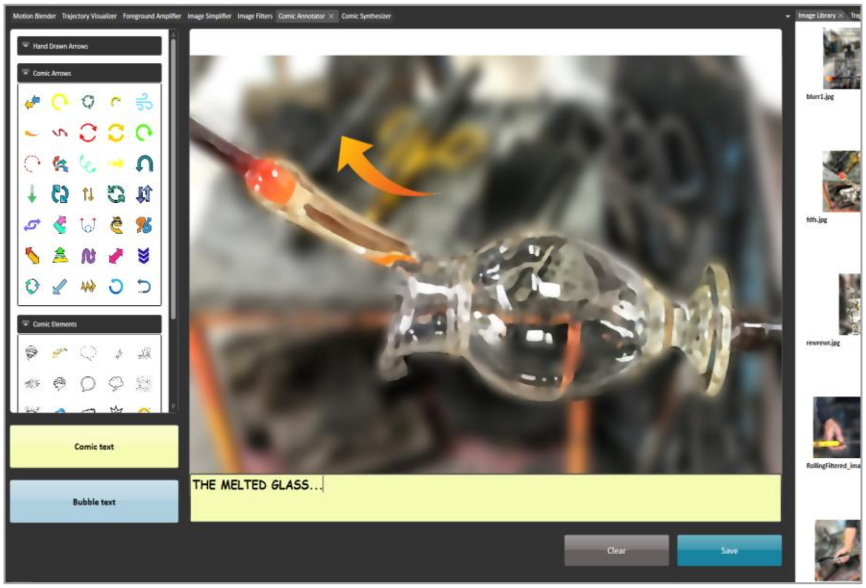


Figure 59. ‘Comic Annotator’ tool – Comic text insertion

In case of speech bubbles, the ‘Comic Elements’ icon set contains a variety of bubble icons that the user can annotate on the top of the input image (Figure 60). If the user clicks the ‘Bubble text’ button, a textbox is inserted on top of the image source, which can be moved by the user inside the dropped speech bubble (Figure 61).

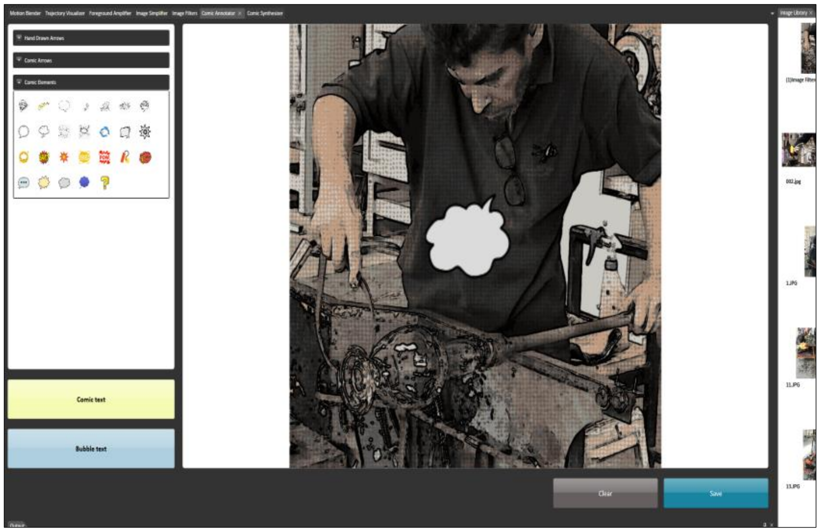


Figure 60. 'Comic Annotator' tool - Bubble icon annotation

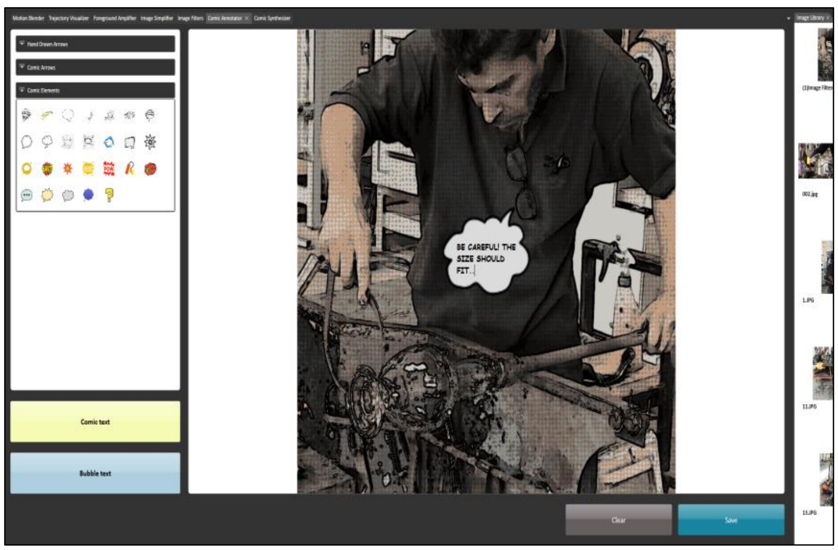


Figure 61. 'Comic Annotator' tool - Bubble text insertion

5.2.7 Comic Synthesizer

The vision of creating artistic visualizations in 2D images is strongly approached in the MotiVo editor via the synthesis of all the edited material exported from the MotiVo tools. The ‘Comic Synthesizer’ tool has been developed for the creation of comic pages providing ready-to-use templates for the users.

- **Input**

  As input, the Comic Synthesizer tool takes multiple image files.

- **Output**

  As output, the Comic Synthesizer tool produces a single comic page.

5.2.7.1 User Interface Key Components

The main components that constitute the user interface of the ‘Comic Synthesizer’ tool are: (1) Image Library, (2) Main area of comic grids and (3) Comic templates area (Figure 62).

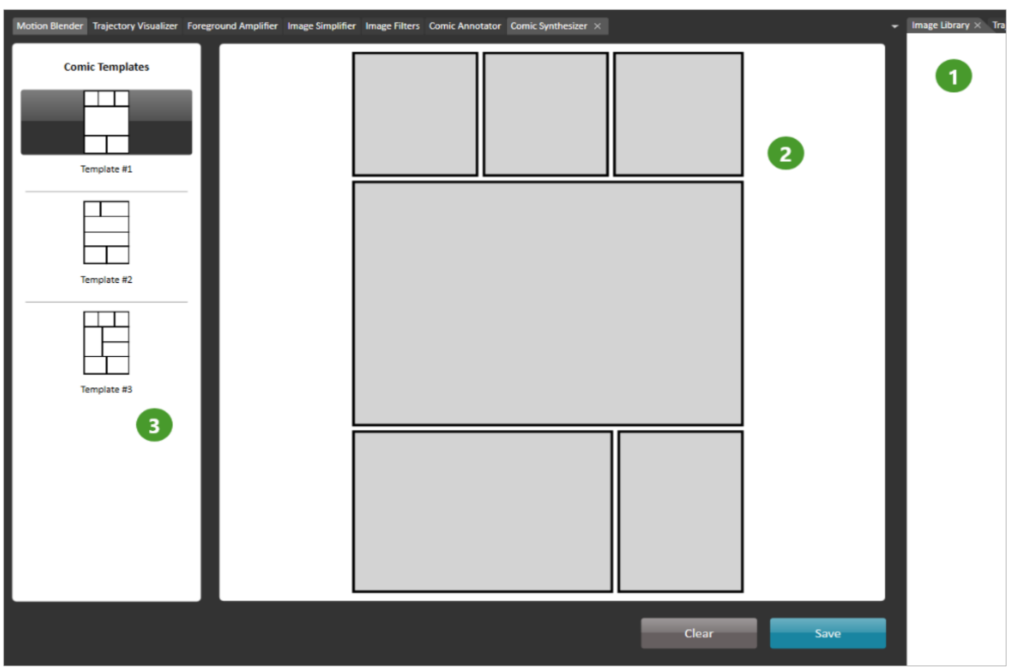


Figure 62. ‘Comic Synthesizer’ tool User Interface

The ‘Image Library’ (i.e. section 1) contains the images required for the comic synthesis. In the ‘Main area of comic grids’ (i.e. section 2) is displayed a gridded comic page where the users can drop edited images.

In the ‘Comic templates area’ (i.e. section 3) there are three (3) available comic templates in the form of buttons that users can select and change the comic page grids in order to create different comic page styles.

5.2.7.2 Use Case

The comic synthesis is performed by selecting the ‘Comic Synthesizer’ tab from the tools bar and then drag-n-drop image files from the ‘Image Library’ inside the grids of the main area of the comic page (Figure 63, Figure 64). All the comic grids should be filled in order the comic page to be ready for saving (Figure 65).

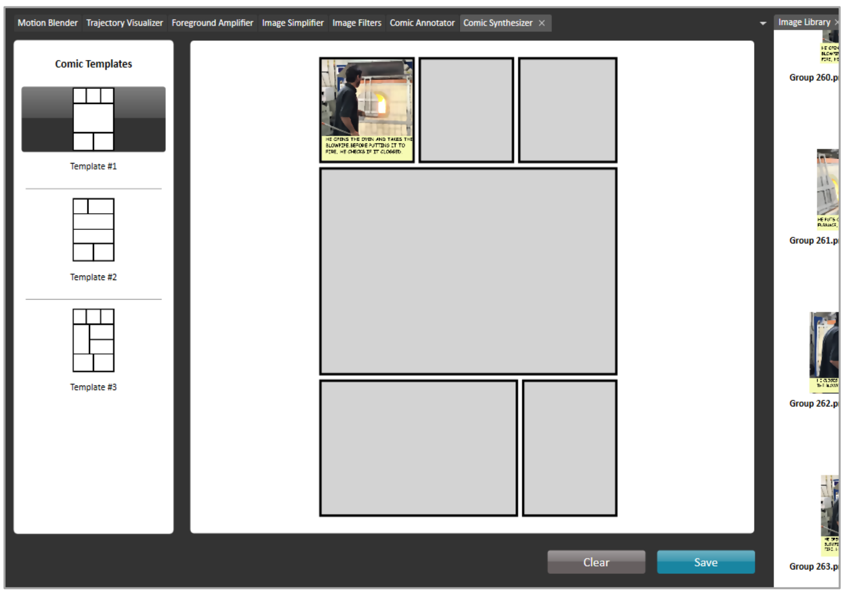


Figure 63. ‘Comic Synthesizer’ tool – First image drop

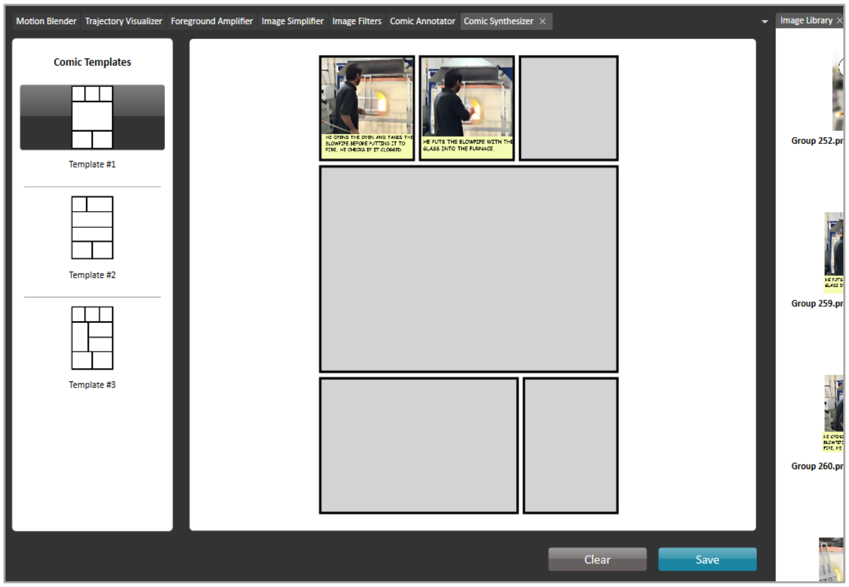


Figure 64. 'Comic Synthesizer' tool – Second image drop

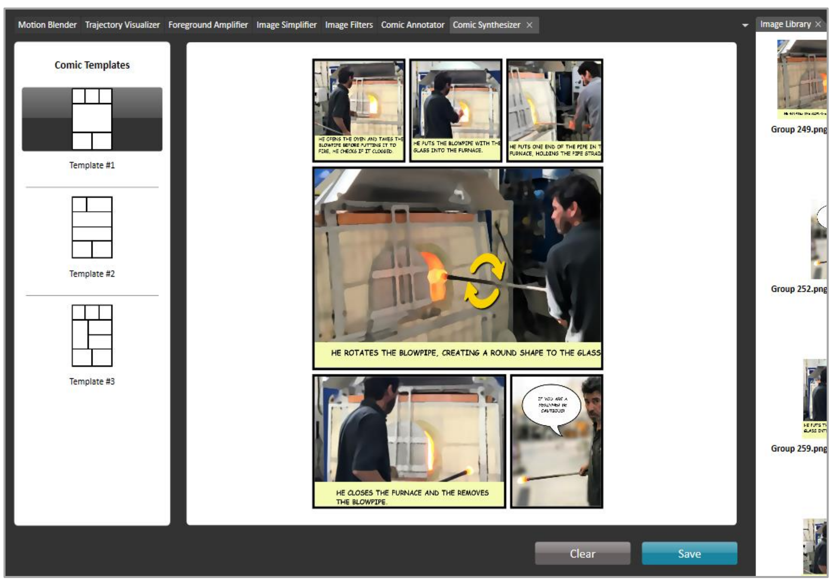


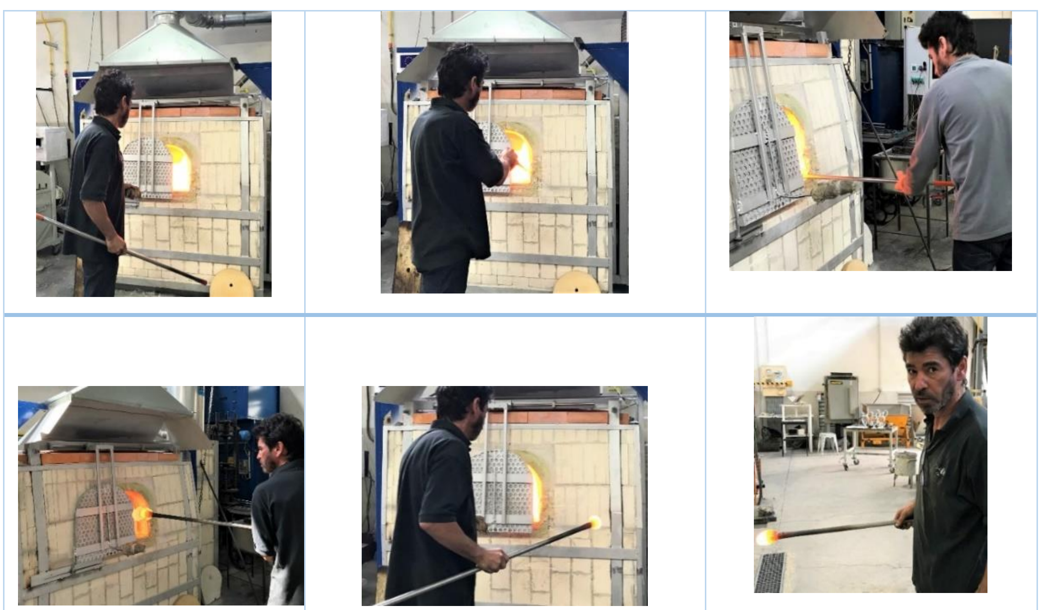
Figure 65. 'Comic Synthesizer' tool – Comic page with all grids filled

5.3 The Comic Book Use Case

Based on the various artistic results of the MotiVo's tools and inspired by the visual closure techniques used in comics (i.e. 'Moment-to-Moment', 'Action-to-Action'), we combined them and created an illustrated comic book that presents a completed test use of the MotiVo editor regarding the creation of the '*Bontemps carafe*'. The illustrated figures were created using the MotiVo editor's tools and by facilitating key-frames extracted by video recordings of the creation process at CERVAF. This comic book has two (2) versions each one based on different approaches. For the first version of the comic book the three (3) of the MotiVo's tools were used; 'Image Filters', 'Comic Annotator' and 'Comic Synthesizer'. In the second version of the comic we used four (4) of the tools; 'Image Simplifier', 'Foreground Amplifier', 'Comic Annotator' and 'Comic Synthesizer'.

In this next sections, we are going to present step by step, the creation of the first page of the comic book for both of the versions. The first page of the comic book is comprised of six (6) images (Table 8).

Table 8. Comic book - Page 1: Original input images



5.3.1 Comic Book (Version 1)

In the first version of this comic book, we mainly focused on the artistic aspect of the MotiVo's tools.

Step 1. 'Image Filters' tool edit

In this step, all of the images were edited using the 'Image Filters' tool. We used the 'Comic' filter category and specifically the 'Preserve color' filter (Figure 66). The output filtered images are presented in Table 9.

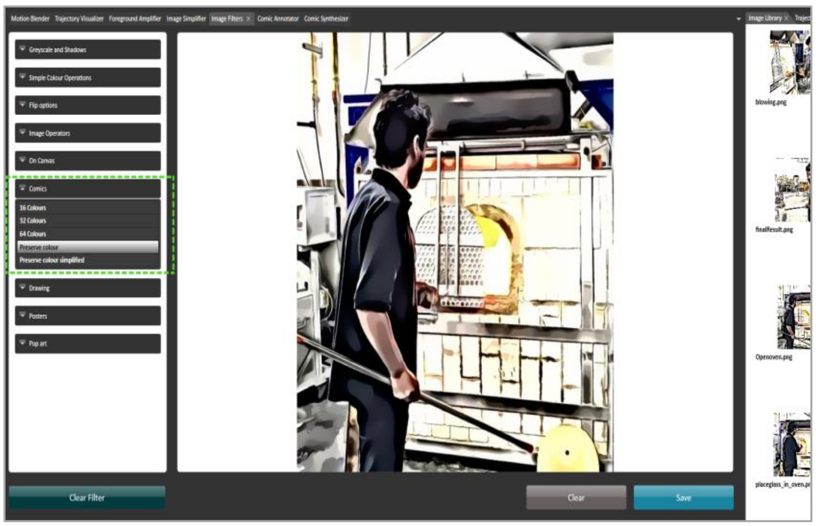


Figure 66. Comic book version 1– Step 1: 'Image Filters' tool editing

Table 9. Comic book version 1 - Step 1: Filtered images produced by 'Image Filters' tool

Step 2. 'Comic Annotator' tool edit

Next, the filtered images were annotated by the 'Comic Annotator' tool. In this step, all the comic and bubble texts were inserted as well as comic elements such as bubbles and arrows. At first, the comic text was inserted into the filtered images (Figure 67). Then, from the left drop-down category of 'Comic Arrows', the green arrow was annotated on the image (Figure 68). The comic annotated images are presented in Table 10.

Figure 67. Comic book version 1 - Step 2: 'Comic Annotator' tool (comic text insertion)

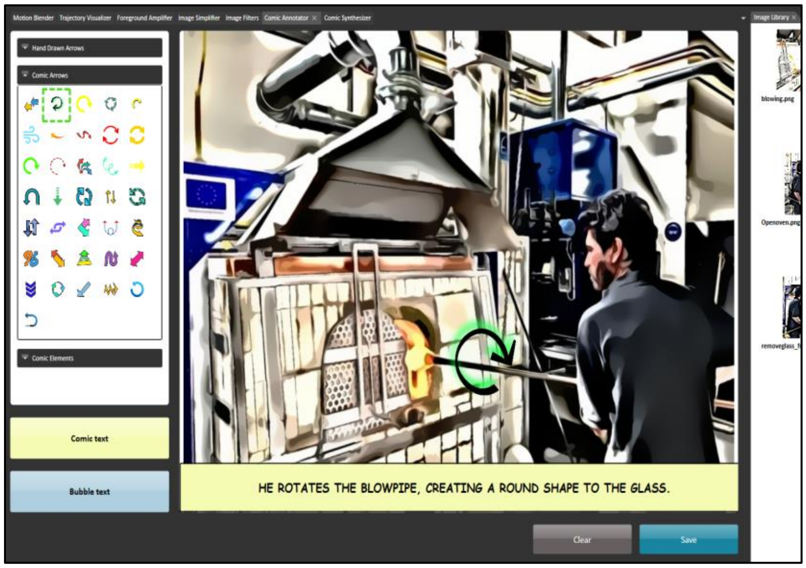
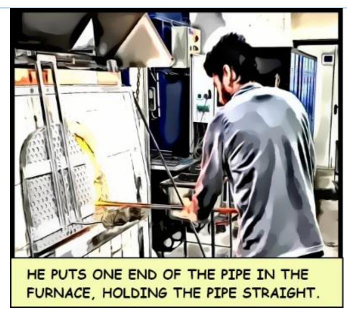


Figure 68. Comic book version 1 - Step 2: 'Comic Annotator' tool (comic arrow annotation)

Table 10. Comic book version 1 - Step 2: Annotated images produced by 'Comic Annotator' tool

Step 3. 'Comic Synthesizer' tool edit

In this step, we created the entire comic page using the 'Comic Synthesizer' tool. The first template out of the three available ones was used. The final result is presented in Figure 69 where all the comic page grids have been filled. The full first version of the comic in presented on Appendix A.

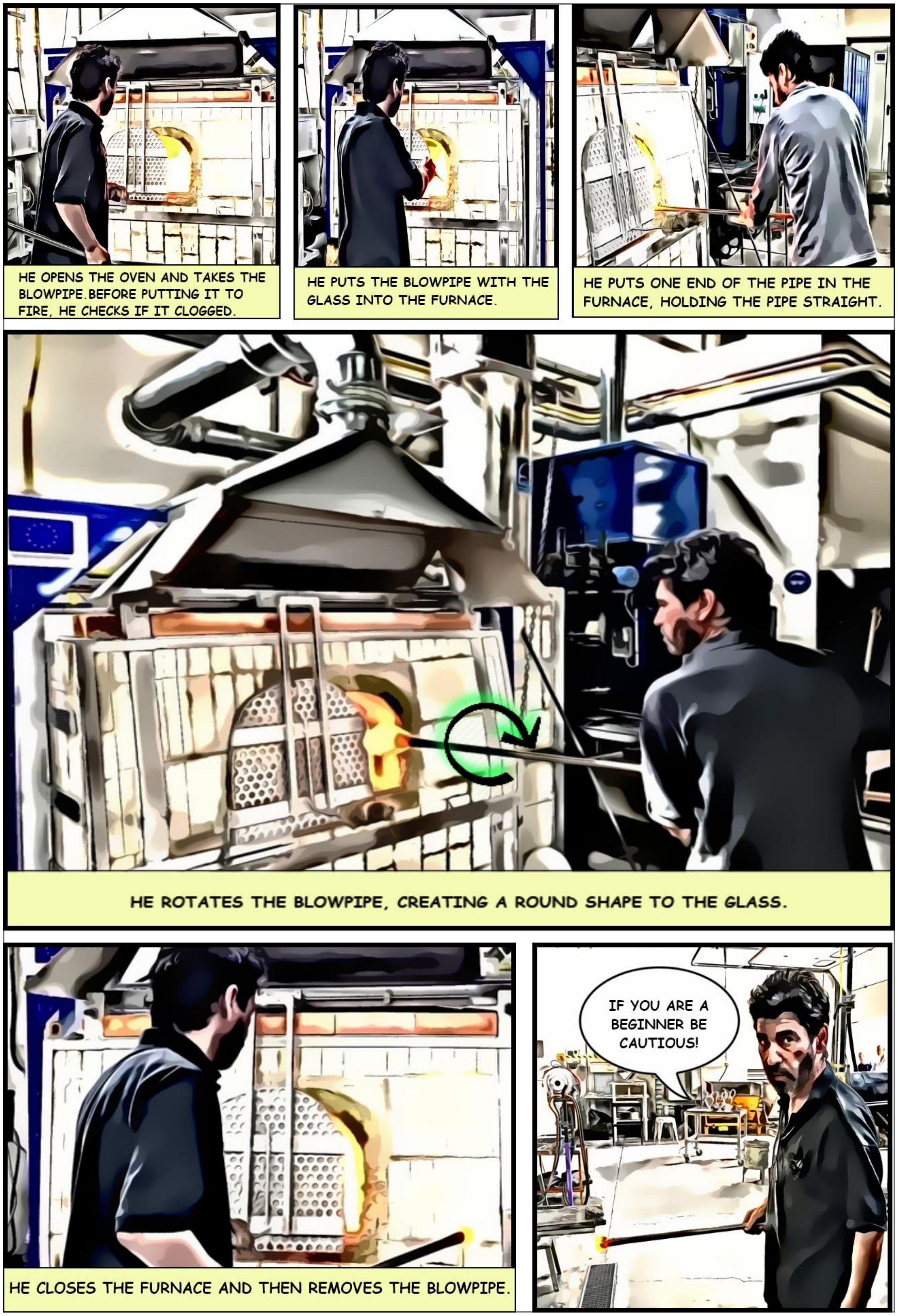


Figure 69. Comic book version 1 – Page 1

5.3.2 Comic Book (Version 2)

In the second version of this comic book, apart from producing an artistic visualization result, focus was given also to the amplification of the main actions on each scene.

Step 1. ‘Image Simplifier’ tool edit

In this step, all of the images were simplified to fit the comic-style objective using the ‘Image Simplifier’ tool (Figure 70). The output simplified images are presented in Table 11.

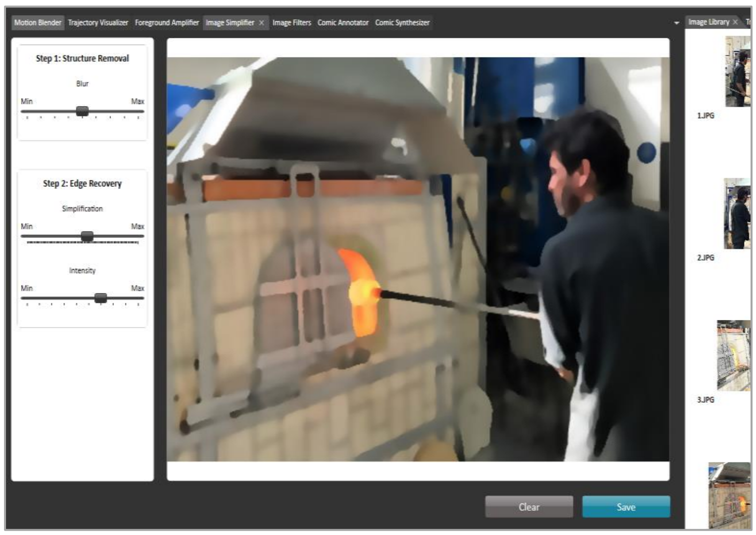
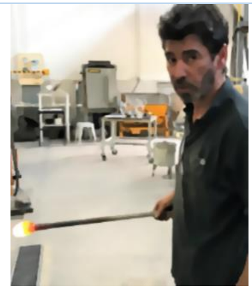


Figure 70. Comic book version 2 – Step 1: ‘Image Simplifier tool editing

Table 11. Comic book version 2 - Step 1: Simplified images produced by ‘Image Simplifier’ tool



Step 3. 'Foreground Amplifier' tool edit

In this step, the 'Foreground Amplifier' tool is used for the amplification of the last image. Firstly, we created a rectangle around the foreground area and defined which part of the scene to amplify (Figure 71). By clicking on the 'Cut rectangle' button on the left panel, the generated result was updated in the main drop image area (Figure 72). Then, by using the foreground and background brushes we further defined these areas as the output result is still not perfectly segmented; the foreground area was drawn with white strokes and the background area with blue strokes (Figure 73). By clicking the 'Cut mask' button the foreground scene was precisely segmented (Figure 74). Finally, from the 'Background Options', the 'Blur' background option was selected. The result is shown in Figure 75.

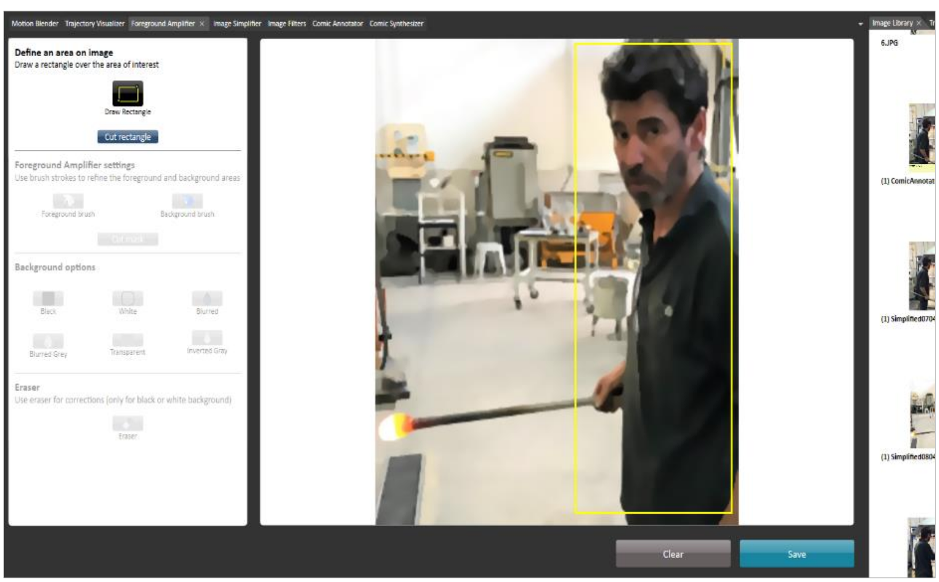


Figure 71. Comic book version 2 – Step 2: 'Foreground Amplifier' tool

(draw rectangle)

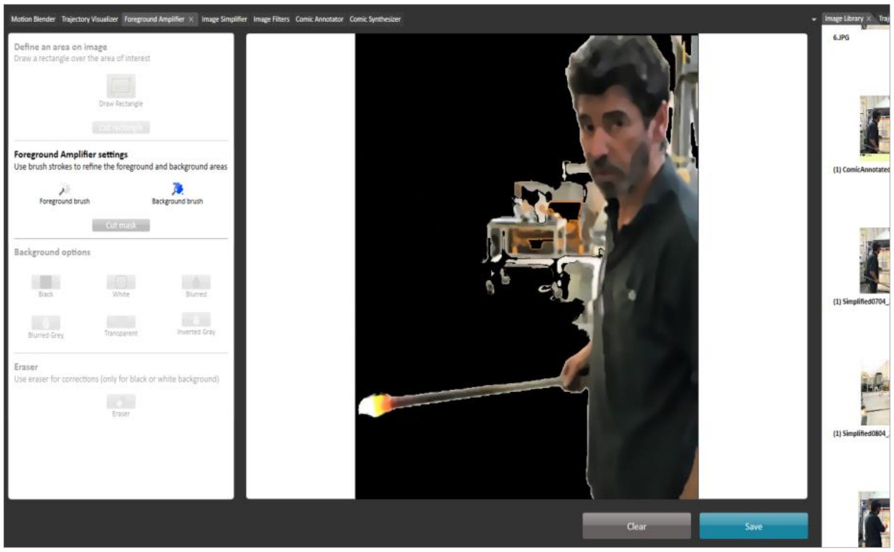


Figure 72. Comic book version 2 – Step 2: 'Foreground Amplifier' tool

('cut rectangle' result)

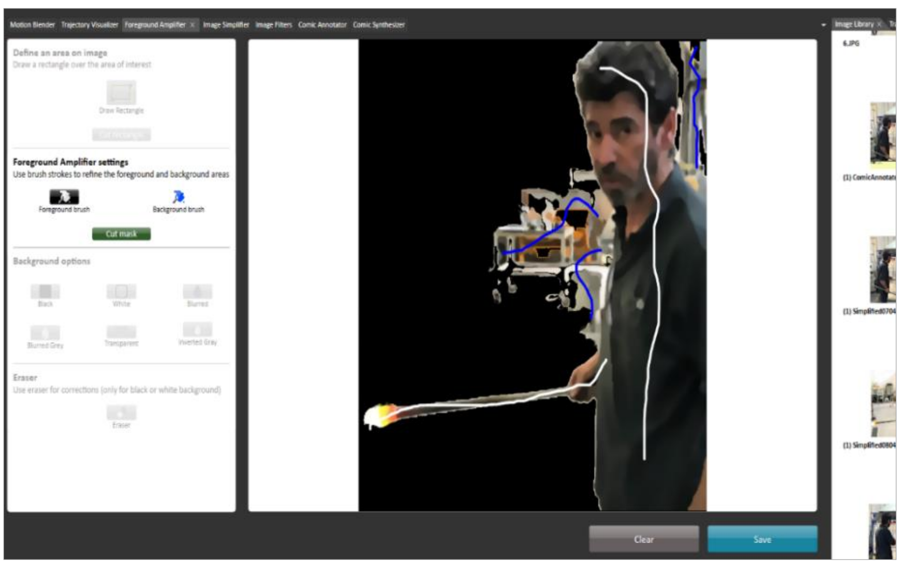


Figure 73. Comic book version 2 – Step 2: 'Foreground Amplifier' tool

(foreground / background strokes)

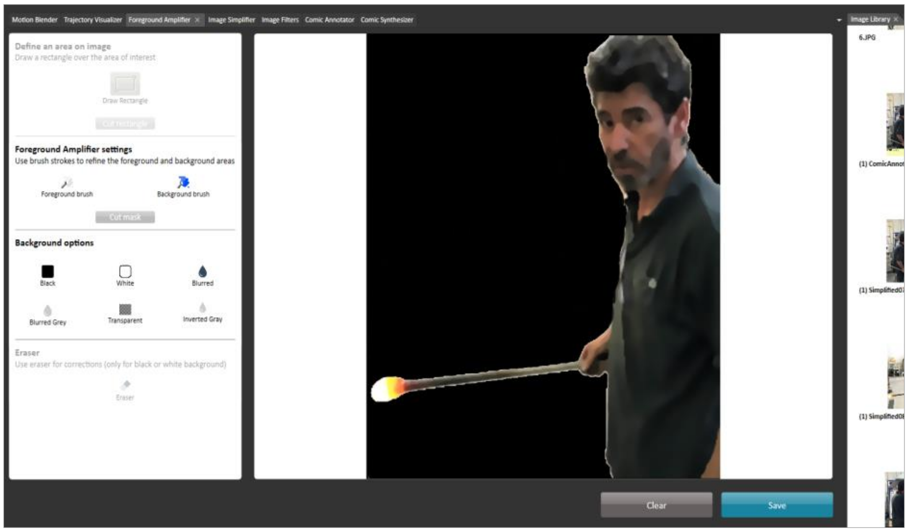


Figure 74. Comic book version 2 – Step 2: 'Foreground Amplifier' tool

('cut mask' result)

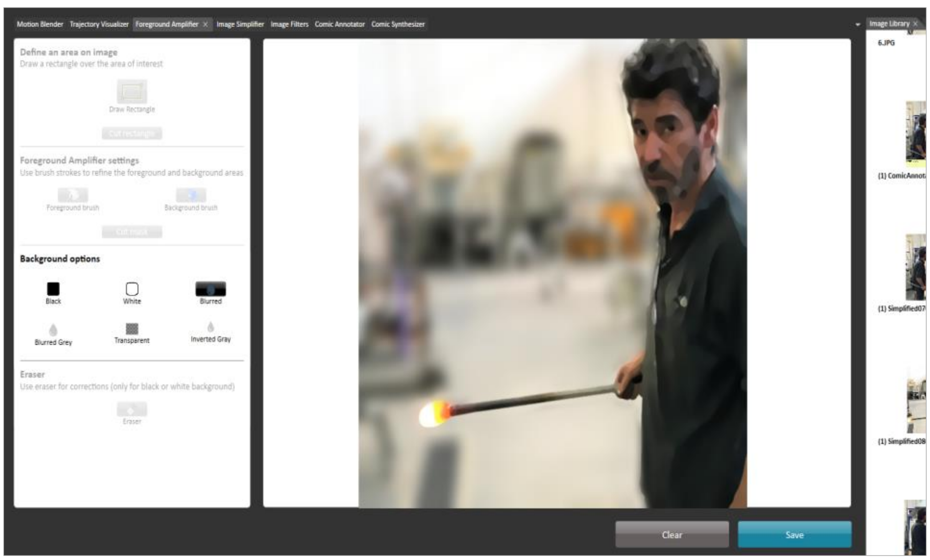


Figure 75. Comic book version 2 – Step 2: 'Foreground Amplifier' tool

(Blurred background)

Step 3. 'Comic Annotator' edit

In this step, the comic texts as well as the comic elements were annotated on the image. For the example of the last image of the first page, a bubble speech icon was added and inside it a bubble text (Figure 76). All the annotated images of the first comic page are presented in Table 12.

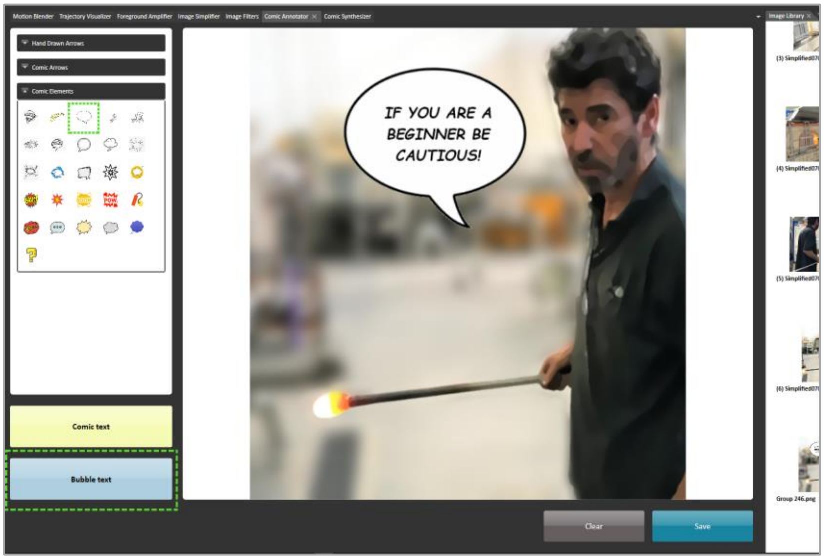
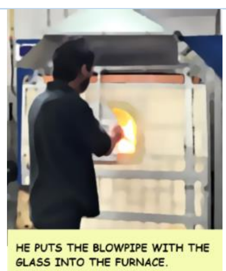
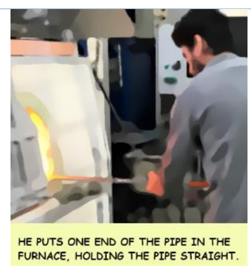
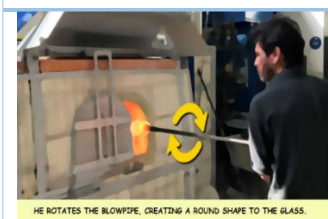


Figure 76. Comic book version 2 – Step 3: 'Comic Annotator' tool

(Bubble icon and text insert)

Table 12. Comic book version 2 - Step 2: Annotated images produced by 'Comic Annotator' tool

Step 4. 'Comic Synthesizer' tool edit

In this step, the entire comic page was created via the 'Comic Synthesizer' tool. The first template out of the three available ones was used. The final result is presented in where all the comic page grids have been filled and saved (Figure 77). The full second version of the comic in presented on Appendix B .

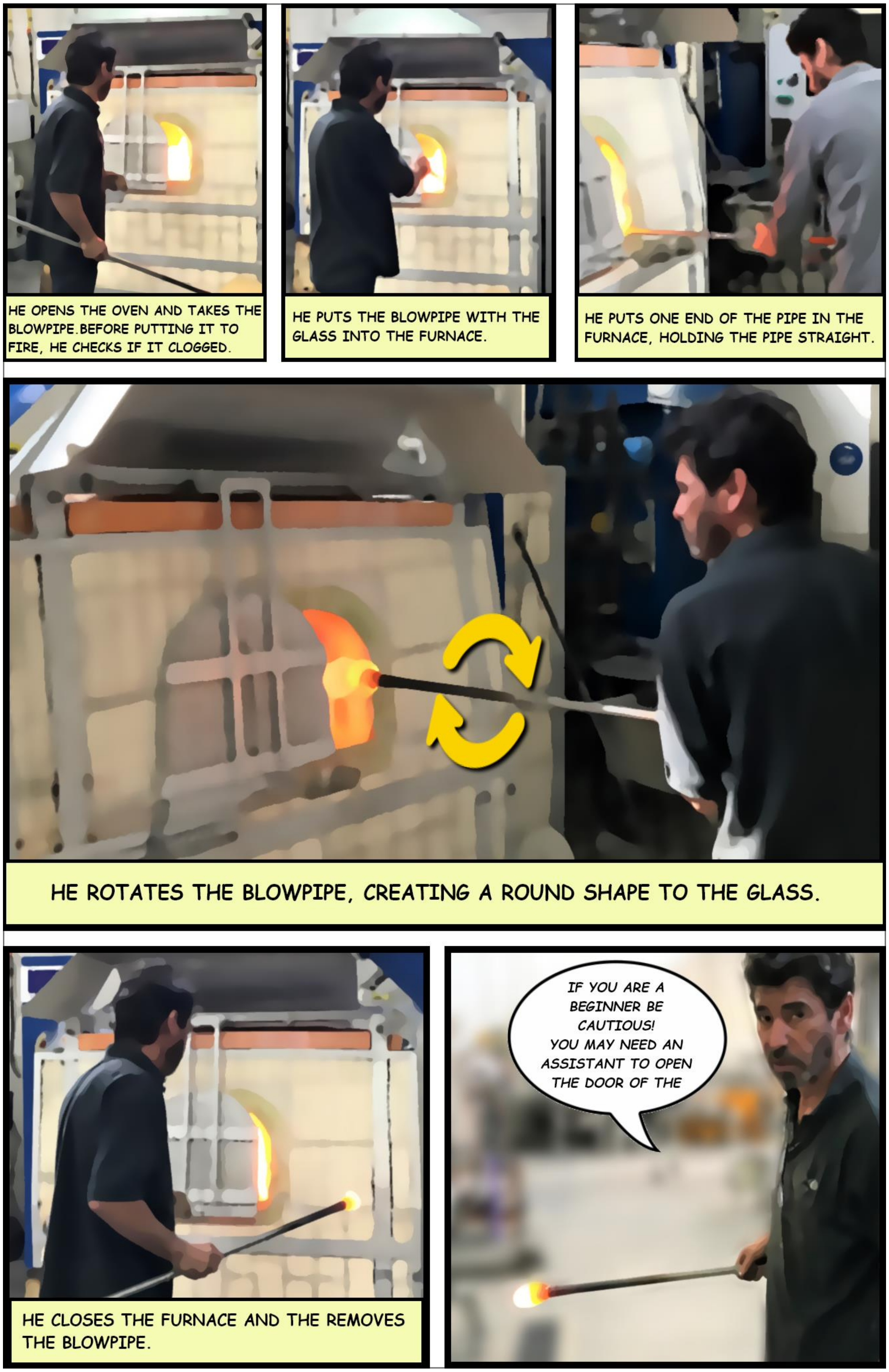


Figure 77. Comic book version 2 – Page 1

5.4 Guidelines

Out of the outcomes of experimenting with the MotiVo's tools and during the examined use case of the comics' creation, a set of guidelines was formed to address the needs of each tool for users to have a better experience with the MotiVo editor.

5.4.1.1 Motion Blender Guidelines

After experimenting with different input files and weight values for each image frame, we concluded that camera functionalities are playing a significant role in the final product.

Guideline 1. The Motion Blender tool produces clearer results in stable camera focus in 2D space.

Guideline 2. Input image sources should have the same size for a successful blended result.

Guideline 3. On a non-stable camera, the output is slightly blurred and disoriented without aesthetically displaying the exact activity. In this case, several experiments may be required from the user side to acquire a sufficient visualization.

Guideline 4. When working with a moving camera select a frame of reference where the camera is static and another one with a lot of changes happening in the initial scene. This will improve the blending quality.

5.4.1.2 Foreground Amplifier Guidelines

Guideline 1. The entire foreground area should be inside the drawn rectangle and if a background spot is inside that area, the background brush will remove it.

Guideline 2. Foreground/Background brushes are used as directives of the area – small strokes on the corresponding areas are enough for the extraction.

Guideline 3. For small resolution input images, the foreground/background extraction process is faster.

5.4.1.3 Image Simplifier Guidelines

Guideline 1. The 'Blur' value (Step 1*)* should be great enough for more intense simplifications.

Guideline 2. For small resolution input images the simplification process is faster.

5.4.1.4 Image Filters Guidelines

Guideline 1. The Image filters are a powerful tool for post and pre-processing results. Experimentation gives an overview of the potential outcomes.

Guideline 2. When a motion blended output has low contrast, image filters facilitate the amplification of that input and thus get better visualization results.

5.4.1.5 Trajectory Visualizer Guidelines

Guideline 1. The use of 2D trajectories can sometimes be non-representative. The total number of points (x, y) should be great enough for the trajectory visualization to be precise. Even though we use Bezier splines to design curves, the input files content should be a good starting point for the Trajectory Visualizer.

Guideline 2. It is suggested the input image be a blended output from the 'Motion Blender' tool as it makes sense for the trajectory depiction.

5.4.1.6 Comic Annotator Guidelines

Guideline 1. Depending on the nature and style of the annotated image, users should choose graphic elements of a similar style to fit the context.

Guideline 2. For comic style output, annotations could be comic styled or even multi-coloured whereas in the case of simple and minimal images, 'Hand Drawn Elements' are the appropriate ones.

Guideline 3. The selected annotation stickers should have enough contrast to the image background to be visible.

Guideline 4. Users are advised to adjust the projection on the annotation stickers so as to indicate the depth and direction of motion.

5.4.1.7 Comic Synthesizer Guidelines

Guideline 1. It is suggested for this tool to take as inputs edited images (i.e. filtered, simplified, annotated) instead of selecting original images for the synthesis of the comic and then try to edit each one.

Guideline 2. The selection of the input image for each grid should be guided by the aspect ratio (e.g., square grids should take similar image sizes).

Guideline 3. For small grid areas, images should be highly simplified.

Chapter 6

Evaluation

The evaluation of the MotiVo editor was conducted at the Human-Computer Interaction Laboratory (HCI) of the Institute of Computer Science of the Foundation for Research and Technology - Hellas (ICS-FORTH). Following the iterative design approach [105] of the applied UCD methodology [95], four (4) preliminary expert-based evaluations were conducted, one after each design iteration. These evaluations were conducted by an HCI usability and interaction expert and/or a technology domain expert using the cognitive walkthrough inspection technique [106] (Section 6.1). For the final version of the application, an expert-based evaluation in the form of a Heuristics evaluation [107] was conducted by three (3) HCI experts and two (2) technical domain experts with great experience in evaluations on interaction systems (Section 6.2). The goal of the evaluations was to identify any potential usability issues regarding the concept, the application features, and UI interaction elements [108] in each produced application version.

6.1 Evaluation Methods

There are a few inspection techniques available for expert-based evaluation, but the most common ones and the ones that were used for the evaluation of the MotiVo editor are cognitive walkthroughs and heuristics evaluation.

Heuristics evaluation [107] is a commonly used method in the HCI field especially in early design iterations because it is effective, quick to produce results, and does not require many resources. In addition, it can, in principle, take into account a wider range of users and tasks than user-based evaluation and assess if the application or system satisfies user requirements. During the heuristic evaluation the inspection is ideally conducted by HCI usability experts who base their judgement on prior experiences and knowledge of common human factors and ergonomics guidelines, principles, and standards. Heuristics evaluations, as well as cognitive walkthroughs, can also be performed by technology domain experts with experience in common design practices in their field of expertise.

In cognitive walkthroughs [106], the expert examines the working or non-working prototype, through the eyes of the user, performing typical tasks and identifying areas in the design or the functionality that could potentially cause confusion or user errors

Both described evaluation methods produce a list of identified issues that can potentially affect the user-system interaction experience. The results of each design and evaluation iteration for the MotiVo editor are presented in the following paragraphs.

6.1.1 First Version Iteration

In the first (1st) version of MotiVo editor, the ‘Motion Blender’ and the ‘Trajectory Visualizer’ tools were incorporated in the UI of the main system (Figure 78, Figure 79). Using the ‘DevZest WPF’ dockable library, a toolbar with the basic functionalities of managing a project (create, save, open) and adjusting the Window environment of the application (themes, view, window) were introduced (Figure 80). Furthermore, a horizontal tab menu was added to host each tool’s access point for the users. This design iteration was inspected by one (1) technology domain expert from FORTH, to ensure that the functionality covered the requirements for these tools and that the overall application UI environment conformed to practices followed in common image editing applications and to the UI dockable paradigm. This inspection produced a list of reported issues, presented in Table 13.

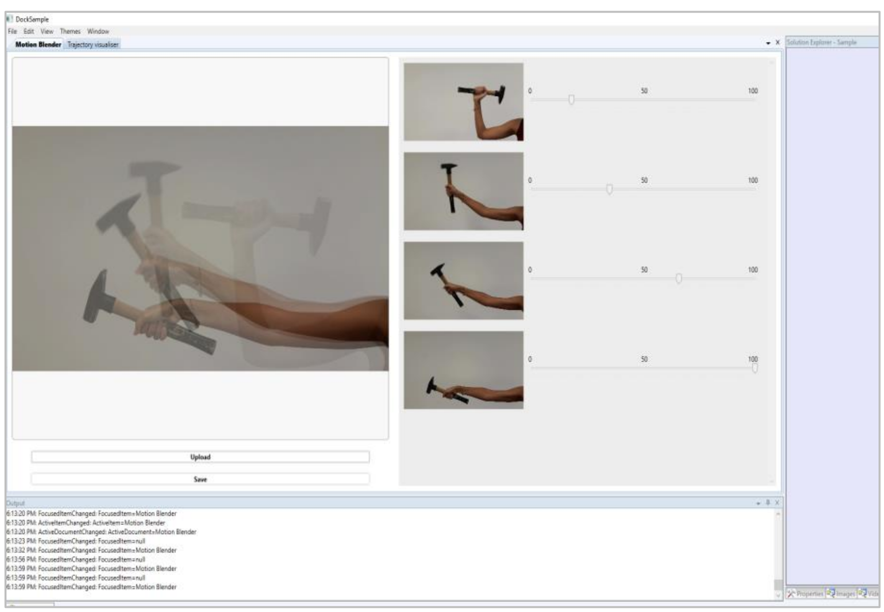


Figure 78. MotiVo Editor Version 1 – ‘Motion Blender’ tool.

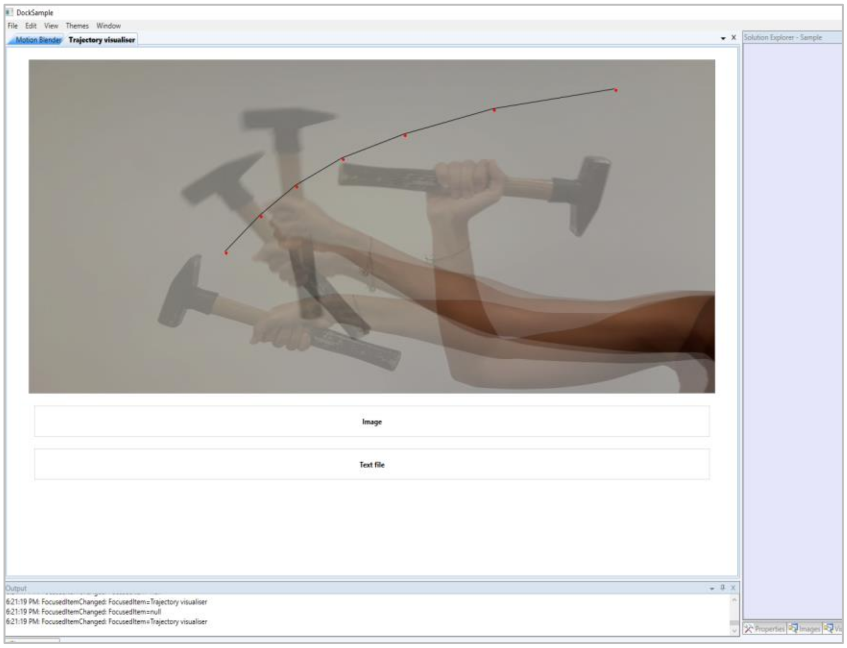


Figure 79. MotiVo Editor Version 1 – 'Trajectory Visualizer' tool.

Applied trajectory points on blended image.

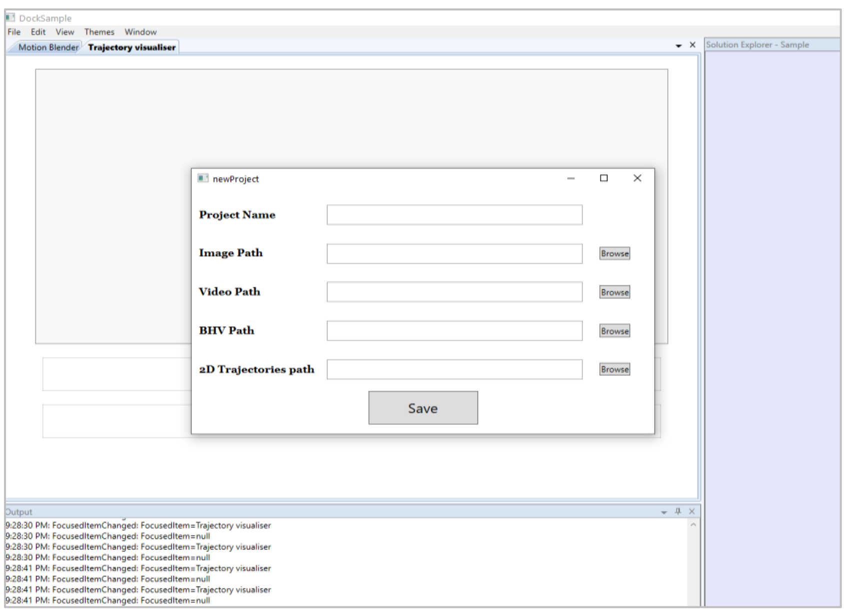


Figure 80. MotiVo Editor Version 1 – Implemented Window functionality to create projects

Table 13. First expert-based evaluation – List of reported issues

Issue #1	The digital assets (i.e., images and trajectory files) contained in the directory paths defined by the user during the creation of a project should appear on the right panel of the system 'Solution explorer', under its respective tab.
Issue #2	Incorporate drag and drop functionality for easy transfer of an asset from the right panel to the working space of any tool. Remove the functionality of importing of each asset individually.
Issue #3	The labels of the tab menu should change to 'Motion Blender' and 'Trajectory Visualizer'.
Issue #4	Use the third variation of the UI shell theme (i.e., Dark Theme) and incorporate the Mingei logo.
Issue #5	In the 'Trajectory Visualizer' tool, the annotated trajectory should include more co-ordinate points to depict the motion path more accurately.

Based on the above reported issues, the following adjustments to the design and the functionality were resolved leading to the second (2^{nd}) version of the editor:

- The libraries of the assets were loaded on the right panel under their respective Tab, instead of being imported inside the working space of each tool (Figure 81).
- The third theme (Dark theme) of 'DevZest WPF' dockable library was applied as well as the Mingei logo (Figure 81).
- Drag and drop functionalities were added, which allows the user to select individual assets from the directory (right container) and place them in the working space.
- Each tool tab was labeled appropriately.
- 'Foreground Amplifier' tool UI was designed according to its main functionality – rectangle cut and mask cut (Figure 82, Figure 83).

Furthermore, three more tools were embedded in the main system; 'Image Filters', 'Comic Annotator' (Figure 84) and 'Foreground Amplifier' tool (Figure 82, Figure 83). The UI design format of these three tools followed the same minimal design of the first tools.

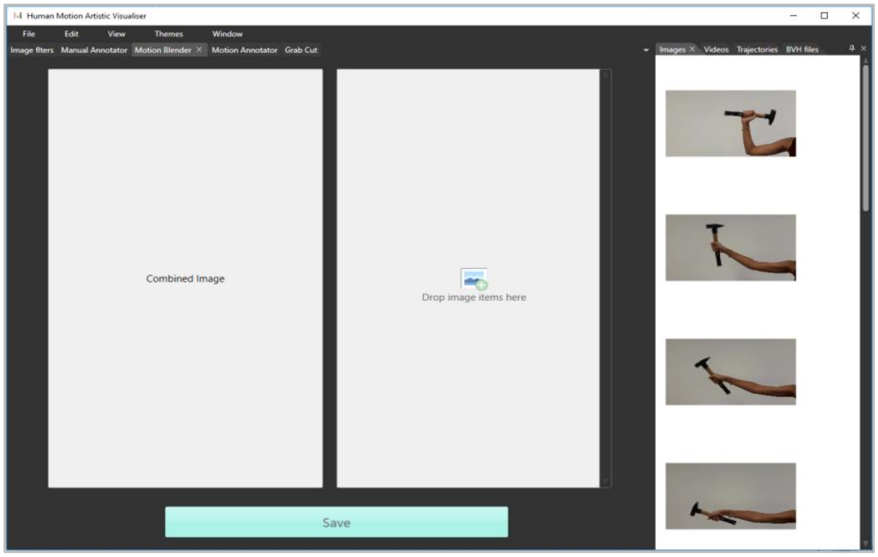


Figure 81. MotiVo Editor Version 2 - Mingei logo (left top corner) and

Dark Theme of shell applied.

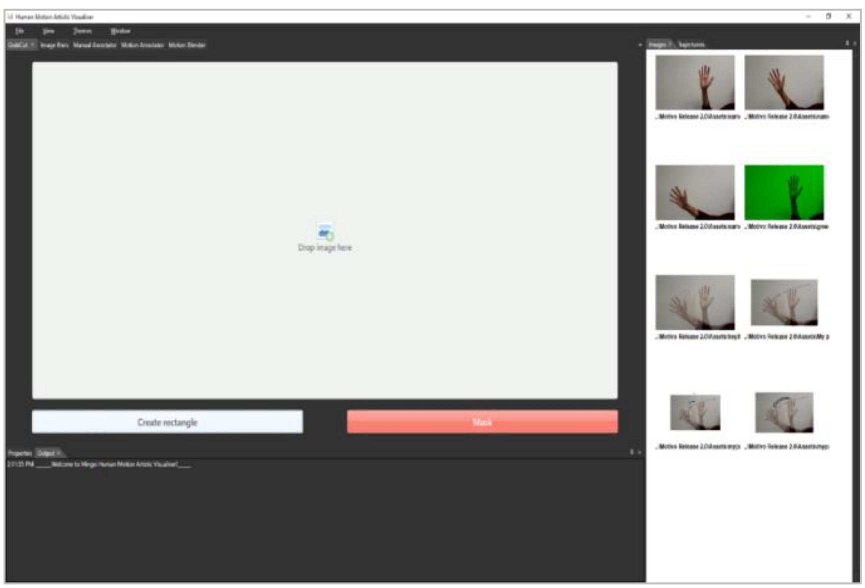


Figure 82. MotiVo Editor Version 2 – Foreground Amplifier tool

(Start screen)

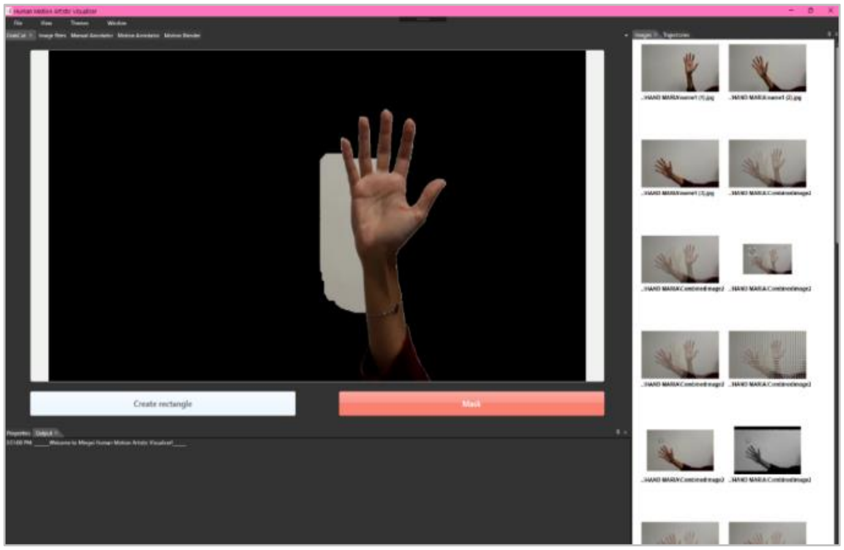


Figure 83. MotiVo Editor Version 2 – Foreground Amplifier tool

('Create rectangle' functionality)

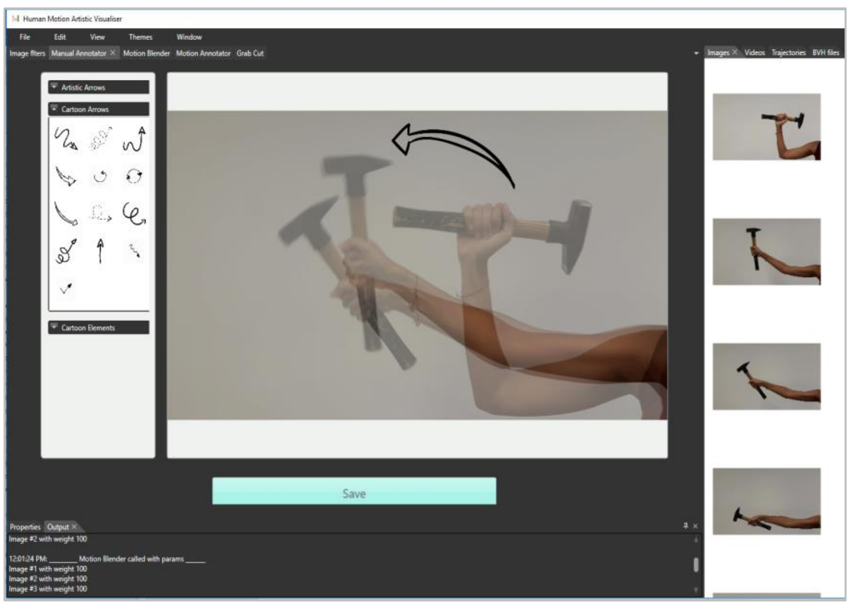


Figure 84. MotiVo Editor Version 2 - 'Comic Annotator' tool

6.1.2 Second Version Iteration

The second version of the MotiVo editor went through a cognitive walkthrough by two (2) HCI usability and interaction experts from FORTH. The results of the second expert-based evaluation are presented in Table 14. The reported issues were directly resolved, leading MotiVo editor to the next version.

Table 14. Second expert-based evaluation – List of reported issues

Issue #1	When creating a project, it is mandatory for the users to add as input all the directory paths (images, videos, trajectories, BVH files) even if they do not want to use all of these asset types.
Issue #2	The filenames of the loaded assets (images, trajectories etc.) are not displayed on the library area.
Issue #3	When the 'Save' button is clicked, the users cannot rename the edited asset – it is automatically given by the system. Moreover, there is no confirmation message that the action was completed successfully.
Issue #4	In the 'Motion Blender' tool, the sliders for changing the contrast weight of each image does not have a label to explain their functionality.
Issue #5	Clear option is missing from all tools – users cannot clear the working space, instead they have to close the entire application and run it again.
Issue #6	The current version does not support functionality to close a project and open another one. Thus, the user has to close the entire application in order to open a different project, which is cumbersome for the user.
Issue #7	There is no 'Undo' mechanism to allow a user to go one step back during the application of any of the supported tool actions.
Issue #8	Help documentation is missing from the system. The documentation should provide information on how to use each tool and descriptions of the supported functionality.

Based on the above reported issues, in the third (3rd) version of MotiVo editor, the following issues were resolved:

- For the creation of a project, only the image file path is mandatory to be filled whereas the trajectory is optional. In fact, videos and .BVH assets have been removed from this editor as they are not used by the tools (Figure 85).
- The image filename is now displayed in the respective asset libraries (Figure 86).
- A prompt window for allowing the user to rename the newly produced output file was added, along with a confirmation message, when action is completed (Figure 87, Figure 88).
- In 'Motion Blender' tool, the slider labels were added.
- Functionality was added to 'Clear' the working space of a tool and to 'Close Project' without having to exit the application.

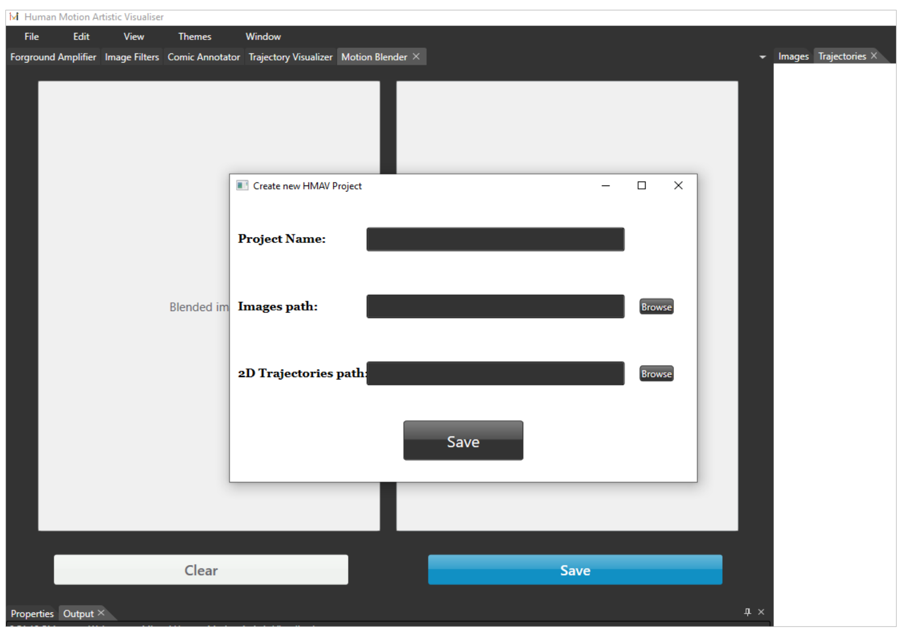


Figure 85. MotiVo Editor Version 3 - Create a new project dialog UI.

Only image path is required to create a project

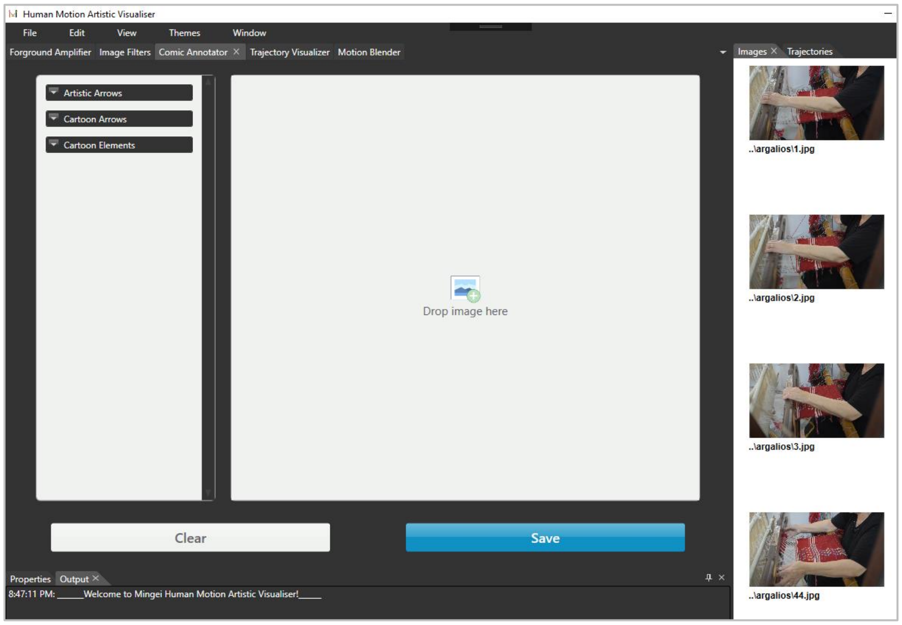


Figure 86. MotiVo Editor Version 3 – Image asset names

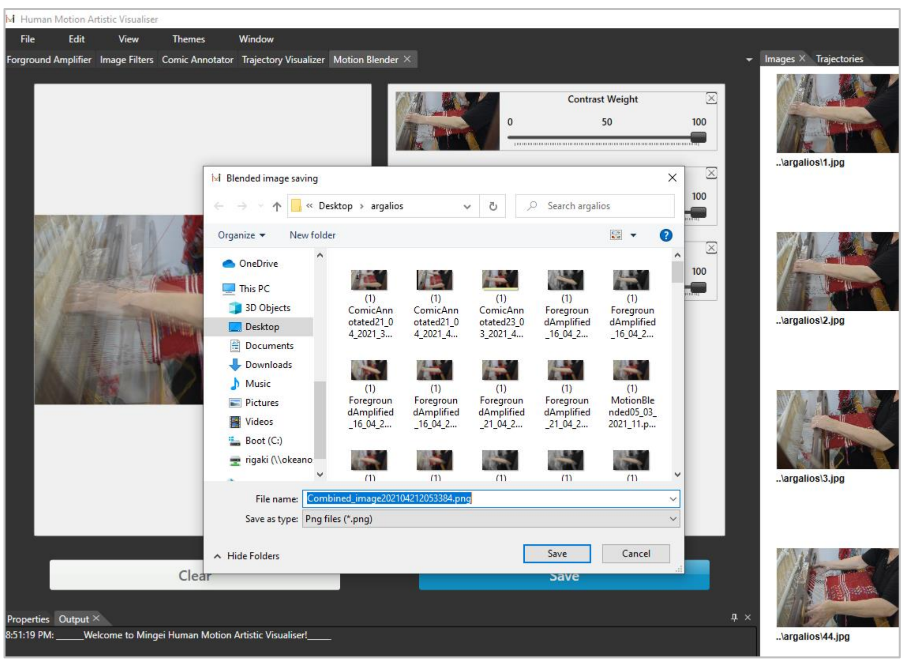


Figure 87. MotiVo Editor Version 3 – Prompt window for renaming saved output

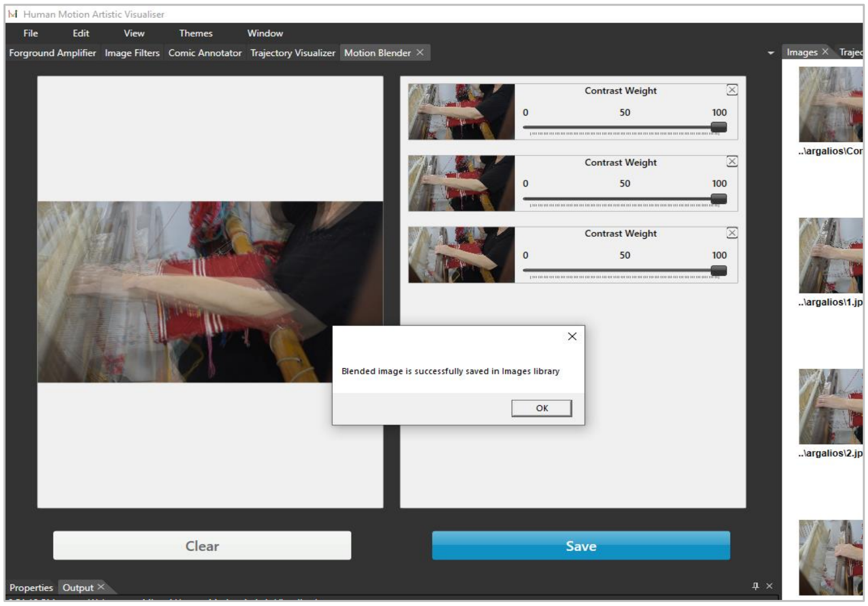


Figure 88. MotiVo Editor Version 3 –Confirmation message on saving

6.1.3 Third Version Iteration

The third (3rd) version of the MotiVo editor went through another cognitive walkthrough evaluation by one (1) HCI usability and interaction expert. The results of this evaluation are presented in Table 15.

Table 15. Third expert-based evaluation – List of reported issues

Issue #1	UI is not consistent across all MotiVo tools.
Issue #2	The assets' (i.e. images, trajectories) name is too long; the entire file path is labelled.
Issue #3	When loading the MotiVo editor, the 'Trajectory Library' tab is by default selected however, all of the tools require Image assets.
Issue #4	When the users open a project there is no evidence of which project has been loaded (the project name is not labelled anywhere).

Issue #5	The ‘Trajectory Visualizer’ tool does not includes the functionality of clearing a specific trajectory file that has been dropped. ‘Clear Button’ clears the entire workspace.
Issue #6	The ‘Foreground Amplifier’ tool is not interactive; users cannot define the foreground area themselves. Also the functionality at this stage is primitive.

Based on the reported issues and recommendations from the third evaluation of MotiVo editor, the following improvements were incorporated leading to the fourth version:

- The UI across all tools was consistently re-designed (Figure 89, Figure 90, Figure 91 and Figure 92).
- Only the file name of the assets is displayed instead of the entire path.
- ‘Image’ library assets tab is by default selected.
- The current loaded project name is displayed on the top of the MotiVo editor window (next to the logo and title of the editor).
- In ‘Trajectory Visualizer’ tool was added the functionality of clearing a specific trajectory file dropped, instead of clearing the entire workspace (i.e., both dropped image and trajectory file) (Figure 90).
- ‘Foreground Amplifier’ tool was re-designed and became more interactive as the users now are able to define the foreground area by drawing the rectangle area as well as use brushes to further indicate the background areas (Figure 91).
- Foreground Amplifier’ tool, background options were also incorporated (Figure 91).

In the fourth version of MotiVo editor, apart from resolving all the reported issues from the third expert-based evaluation, one (1) more tool was embedded in the main system; ‘Image Simplifier’ tool (Figure 93).

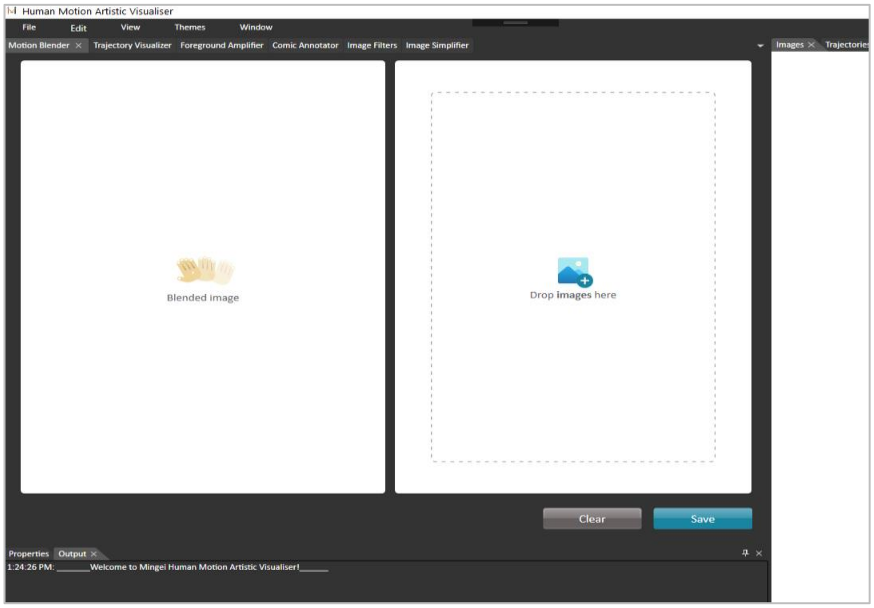


Figure 89. MotiVo Editor Version 4 – 'Motion Blender' tool UI

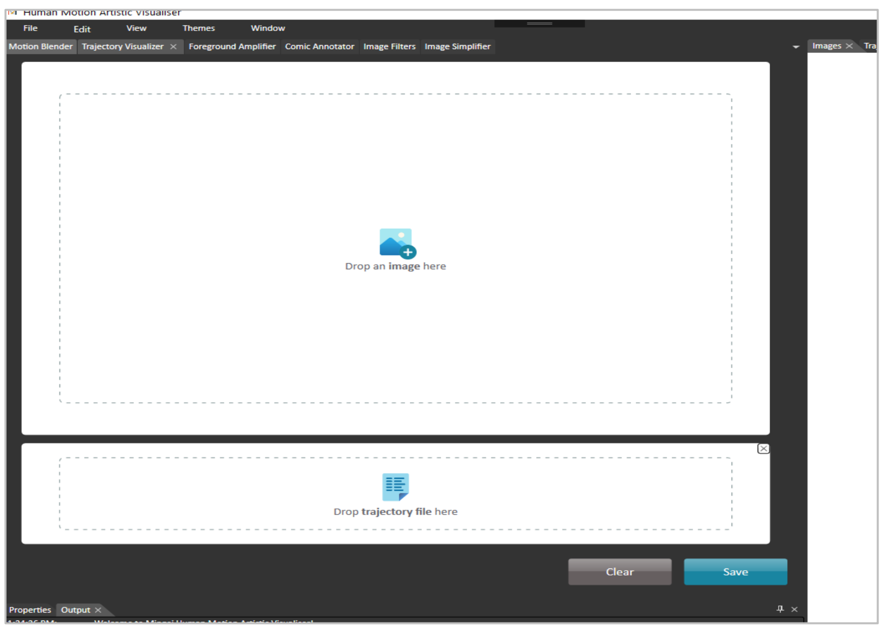


Figure 90. MotiVo Editor Version 4 – 'Trajectory Visualizer' tool UI

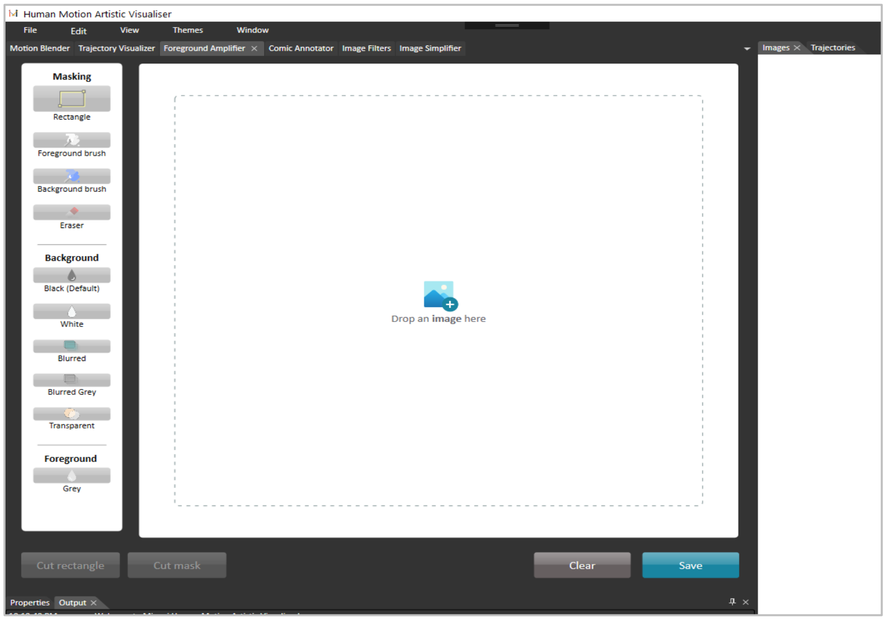


Figure 91. MotiVo Editor Version 4 – Foreground Amplifier tool UI

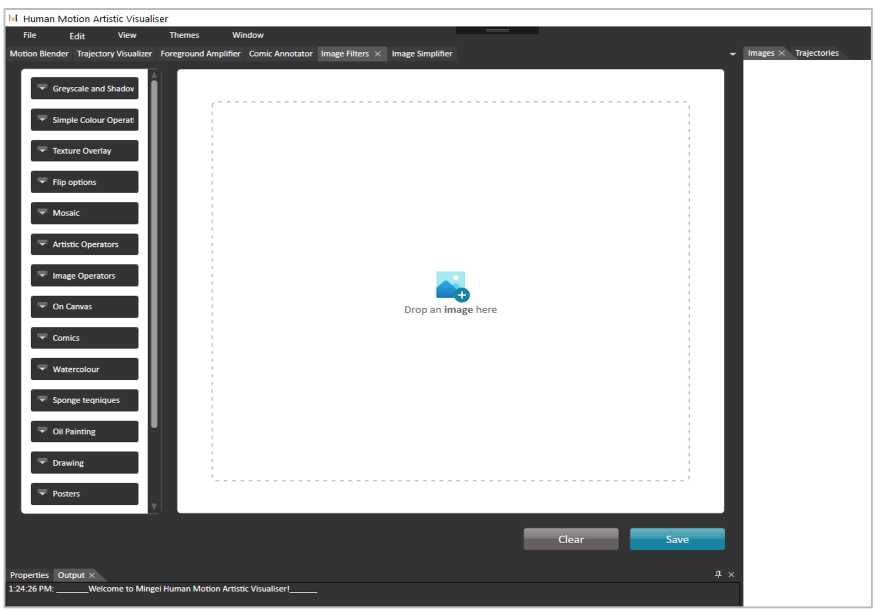


Figure 92. MotiVo Editor Version 4 – Image Filters tool UI

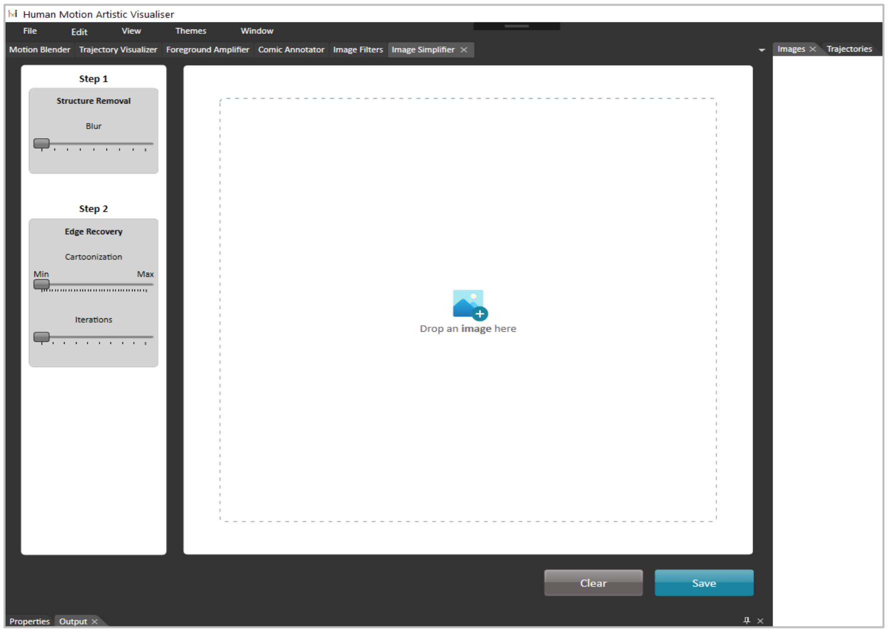


Figure 93. MotiVo Editor Version 4 – Image Simplifier tool UI

6.1.4 Fourth Version Iteration

The fourth version of the MotiVo editor went through another cognitive walkthrough evaluation by one (1) HCI usability and interaction expert and one (1) technology domain expert. The results of this evaluation are presented in Table 16.

Table 16. Fourth expert-based evaluation – List of reported issues

Issue #1	In the 'Trajectory Visualizer' tool, the visualized trajectory on a blended image is not responsive when the window of the application is rescaled.
Issue #2	In the 'Foreground Amplifier' tool, the UI of the left side menu is complicated as the users does not understand which options are enabled and which are disabled.
Issue #3	In the 'Foreground Amplifier' tool, after selecting the rectangle cut option both buttons 'Cut mask' and 'Cut rectangle' are enabled. It is not obvious to the user which one to select and why.
Issue #4	In the 'Foreground Amplifier' tool, the 'Eraser' option can only be applied when the chosen background colors are set to Black or White. It does not work for

	the other background options (i.e. blurred, grey, etc.). This limitation is not obvious to the user.
Issue #5	In the ‘Foreground Amplifier’ tool, each action performed with the eraser option is not automatically saved.
Issue #6	In the ‘Motion Blender’ tool, the produced blended images are saved with the same prefix in the file name, making all files look the same.
Issue #7	In the ‘Image Filters’ tool, there is no ‘clear’ option in order the users to remove a filter when applied.
Issue #8	In the ‘Image Filters’ tool, the filters' list on the left are crowded. Moreover, the selected option cannot be undone unless the user selects another filter.
Issue #9	In the ‘Comic Annotator’ tool, text input for labels is a must! This functionality needs to be supported.
Issue #10	The mental model of ‘Image Simplifier’ tool is also not obvious to the users.

Based on the above reported issues, the following adjustments to the design and the functionality were resolved leading to the fifth version of the editor:

- The UI of the toolbar in the ‘Foreground Amplifier’ tool was re-designed and the available options were enabled and disabled appropriately (Figure 94).
- For each available option in the left toolbar of the ‘Foreground Amplifier’ tool, a specific title was added in order to guide the users (Figure 94).
- In the ‘Image Filters’ tool, some image filters categories were removed as they were not appropriate for MotiVo editor.
- In the ‘Image Filters’ tool, a ‘Clear Filter’ button was added (Figure 95).
- In ‘Motion Blender’ tool, the prefix of each saved image name was differentiated.

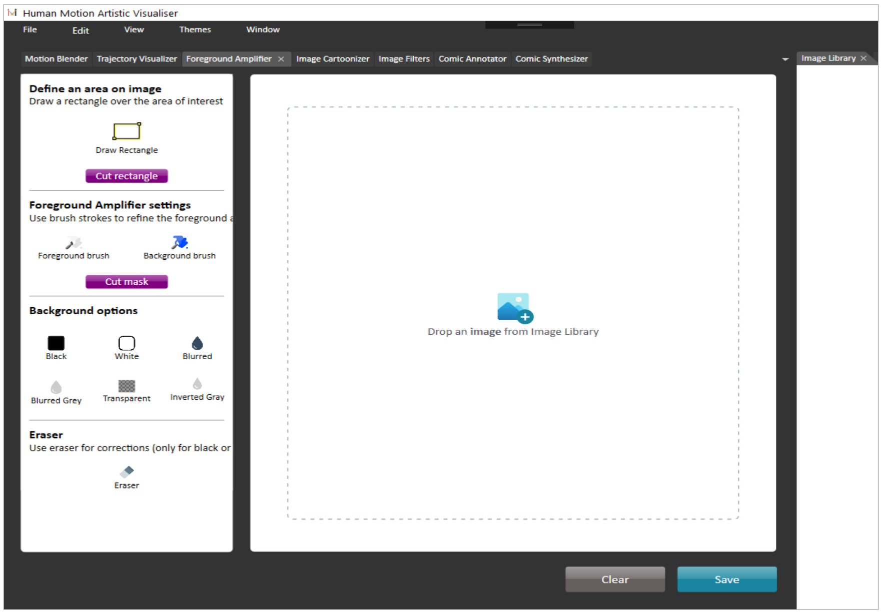


Figure 94. MotiVo Editor Version 5 – Foreground Amplifier tool

Toolbar UI

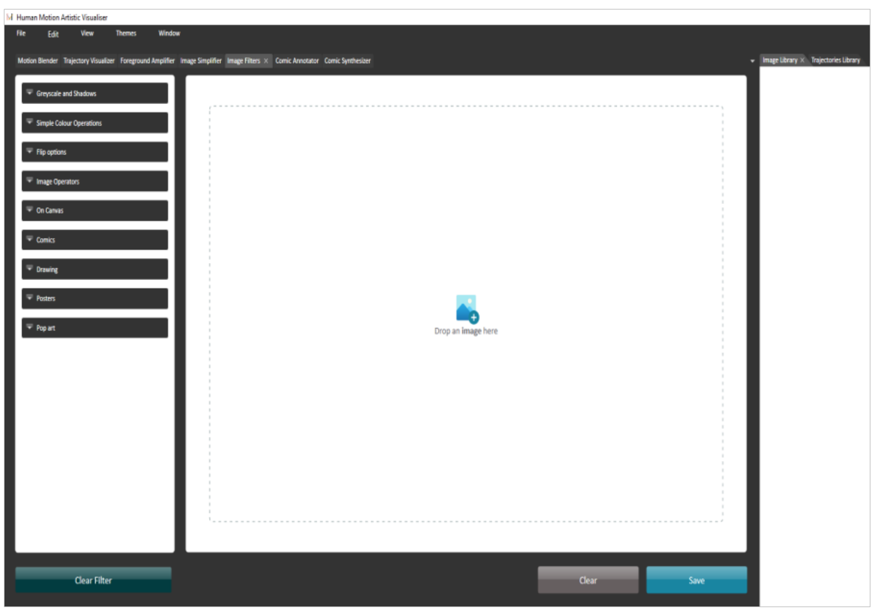


Figure 95. MotiVo Editor Version 5 – Image Filters tool

'Clear Filter' button added

The fifth design iteration of MotiVo editor went through a final expert-based heuristic evaluation for this Thesis. The process as well as the reported issues of the final evaluation are extensively presented in the next section (Section 6.2).

6.2 Final Heuristic Evaluation

The final heuristic evaluation of the MotiVo editor was conducted with the participation of five (5) experts; three (3) HCI usability and interaction experts and two (2) technology domain experts. The main objective of this evaluation process was to extract identified findings by different experts and merge them into a list of usability issues.

6.2.1 Process

During the evaluation process, the experts were invited, one by one, in separate sessions and were introduced to the purpose of this approach, followed by a brief introduction to the system and its functionalities. Since some of the evaluators, three out of five (3 out of 5), did not have any prior familiarity with these tools, they were given a brief description of the main features of MotiVo editor, the context of use of MotiVo tools and their purpose in the context of the Mingei project. The session time for each expert was not restricted.

All experts were prompted to think aloud during each session, since their comments constituted a basis for identifying more easily the usability issues of the system. These issues were afterwards merged into a list, which contained all the unique findings. Each expert was given a copy of this list to provide a severity rating to each issue, according to their preference. These ratings ranged from one (1) to four (4) (decimal ratings were also allowed), with one (1) to be "Aesthetic" issue, two (2) "Minor" issue, three (3) "Major" issue and four (4) "Critical" issue.

The final severity score for each usability issue was obtained by combining and averaging individual scores provided by each expert. Then the identified issues, along with the final severity rating were ranked according to the ease-of-fix scale; zero (0) ("would be extremely easy to fix") to three (3) ("would be difficult to fix"), in order to designate the amount of effort needed for each issue.

6.2.2 Findings

The evaluation on the MotiVo editor revealed forty-seven (47) usability issues, categorized by each tool and sorted based on their severity ratings, as given by the five (5) expert evaluators. Table 17 presents the extracted issues, along with their severity score provided by the experts, the ease-of-fix score as well as their status (i.e. 'Fixed' or 'Future work' (FW)).

Table 17. Heuristic evaluation – List of reported issues (FW=Future Work)

Heuristic Comment	**Severity Score** **1=Aesthetic** **2=Minor** **3=Major** **4=Critical**	**Ease-of-fix Score**	**Status**
General			
When users close any of the MotiVo tool tabs, they cannot reload them.	3	2	Fixed
Users should be able to edit the assigned file paths (i.e. image or trajectory paths) of a created project.	2.8	3	Fixed
The trajectory path should be optional, on the creation of a new project, as users may not use the 'Trajectory Visualizer' tool.	2.8	0	Fixed
Instructions, help pages, or demo videos are needed to describe the functionality of each tool.	2.6	2	FW
Information about which files (images, trajectories) were used to produce an output should be displayed and be available to the users.	2.6	0	Fixed
A 'loading' icon should be displayed during image processing to inform users that a result is being produced.	2.4	0	Fixed
The selected MotiVo tool tab should be highlighted, so the user knows which tool is active.	2.4	0	Fixed
The 'Close' button should be disabled in case there has not been created or opened a project by users.	2.3	1	Fixed

On hover, the 'Image Library' tab should display its directory path.	2.1	1	Fixed
The properties window at the bottom of the UI should be available only on demand.	2.1	0	Fixed
A splash screen with the main options to create or open a project should be displayed, when the user loads the MotiVo editor.	2.1	1	FW
On the creation of a new project, the input fields of the directory paths should not allow typing.	2.1	0	Fixed
The text of all UI elements (i.e. tool tabs, buttons, and description text) should be larger. UI elements are hard to read.	2	0	Fixed
The MotiVo tool tabs should be larger with have more padding between them.	1.7	0	Fixed
In the 'Images Library', when the filename of an image is too long, it should get cut earlier and show '...'. The full name should be displayed on hover.	1.7	1	Fixed
The dialog window on the creation of a new project should not be responsive.	1.7	0	Fixed
Motion Blender tool			
The users did not understand that the saved image is placed in the 'Image Library' despite the pop-up message box.	2.9	2	Fixed
The produced blended images are saved with the same prefix in the file name, making all files look the same.	2.8	0	Fixed
After clicking the 'Clear' button, the warning should say 'All images will be cleared from the window' and not 'deleted' as it is labelled now.	2.7	0	Fixed

When an image file is dropped in the drop images area, its name should also be displayed.	2.7	0	Fixed
The 'Drop images' icon disappears as long as an image file is dropped. It is not clear for the users if another image file is allowed to be dropped.	2.6	1	FW
The 'Clear' and the 'Save' buttons should look disabled when users first open this tool.	2.3	1	FW
Detailed label should be displayed instead of 'Drop images here' (e.g. 'Drag-n-Drop images from the 'Image Library').	2.1	1	Fixed
The addition of an image file in the 'Drop images' container should also be achieved by 'Double-clicking' on its file name from the 'Image Library’.	2.1	2	FW
The 'delete' - x- button on the top right of a dropped image is too small and not obvious.	2	0	Fixed
The text in the warning or saving pop-up windows should become larger, so that user can read it more easily.	1.8	0	Fixed
Trajectory Visualizer tool			
Additional instructions are needed for this tool as the user does not know how to create a trajectory.	2.4	2	FW
Trajectory line sometimes is displayed outside the image border.	2.4	2	FW
Foreground Amplifier tool			
‘Undo’ functionality is not provided on this tool.	3.2	3	FW

Instructions are needed on how to use the foreground/background brushes to define the ROI.	3	2	FW
The ‘reset’ functionality of the original image should be available for the user.	2.8	2	FW
The tool allows users to erase outside the image area. This is not valid.	2.8	2	FW
Users are allowed to draw the rectangle outside the borders of the displayed image.	2.6	2	FW
The eraser should erase in the dimension of the circle cursor.	2.6	2	FW
The drawn rectangle's size should be editable (i.e. user drag on its side to enlarge it, etc.).	2.5	1	FW
The titles of each step, in the left toolbar, are not intuitive.	2.5	0	Fixed
When the eraser tool is clicked, the cursor pointer should be an eraser icon.	2.4	0	Fixed
When applying the transparent background option, the default visual effect (i.e. little white and gray squares) should be depicted as background.	2.4	0	Fixed
When the foreground brush is selected, the mouse pointer should show the same icon, so the user knows which tool is selected.	2.4	0	Fixed
Image Filters tool			
When a filter is selected, all the previously selected filters are still highlighted.	2.9	1	FW
The users are not able to de-select an applied filter (toggle mechanism).	2.8	1	FW

The selected filter category is not highlighted even when the dropdown menu is not expanded.	2.6	2	FW
Mouse scroll is not provided in the filters panel.	2.5	1	Fixed
Comic Annotator tool			
Resizing of annotation elements works backwards (i.e. when user enlarges the width of the box, it increases its height).	2.9	2	FW
The users cannot insert text on top of the dropped images and inside the speech bubbles.	2.9	0	Fixed
The border of the annotation elements should not be visible, after an image is saved and reloaded.	2.7	1	FW
Image Simplifier tool			
'MIN' 'MAX' labels should be added to all UI sliders.	2.5	0	Fixed

The average score of all the issue scores is about two and a half (2.5). Specifically, four (4) issues were ranked as 'Aesthetic' problems (between 1-1.9), thirty-nine (39) as 'Minor' usability problems (between 2-2.9), and the remaining four (4) were ranked as 'Major' issues (between 3-3.9). Zero (0) 'Critical' issues were found during this evaluation.

Most of the issues concerned general interface inconsistencies and aesthetics whereas the major issues were mostly comments about project functionalities (i.e., users can not edit a created project by changing the assigned folder directories).

All the fixed issues are presented in the 'Status' column of Table 17. The rest of the reported usability issues are going to be addressed in the near future. Some of the most important issues that will be addressed in the future include (a) providing an 'undo' functionality across all the MotiVo tools (b) design various ready-to-use templates for alternative purposes and domains and (c) adopt new forms of inputs (e.g. videos). In general twenty-eight (28) out of forty-seven (47) issues were resolved and led our system in its final version for this Thesis. The valuable feedback from the five (5) experts inspired the development of an extra tool; the 'Comic Synthesizer', in order the end-users to synthesize and create collages and artistic material combining the outputs from the MotiVo editor tools.

Chapter 7

Conclusion & Future Work

This Master Thesis was conducted in the context of the Mingei H2020 EU project which supports the vision of representation and preservation of Heritage Crafts due to their importance and value. The main purpose of this research work was to build over centuries of visual art experience and move forward into proposing artistic motion visualization techniques on static media based on human perception. Specifically, inspired by motion visualization techniques used in art, comics, photography and instruction manuals, we marched into implementing the MotiVo editor, a 2D Human Motion Artistic Visualizer that targets on the representation of Heritage Crafts, by assisting in the semi-automatic creation of readable visualizations and movement summarizations. Moreover, it produces artistic image depictions of craft processes and techniques in order to enhance the educational and presentation value of digital content. The MotiVo editor is comprised of seven (7) distinct tools built on top of a plugin-based architecture capable of being extended and enriched in the future. These tools exploit different visualization techniques based on relevant bibliography.

The 'Motion Blender' tool was based on the technique of multiple images used in photography and creates a directional motion effect by blending different key poses of an action using user-specified intensity weights for each one. The 'Foreground Amplifier' tool was inspired by the figure-ground theory thus it amplifies the foreground region of interest as it creates image masks and provides several background options in order to de-emphasize the background scene. Inspired by texture techniques used in art and comics, the 'Image Simplifier' tool simplifies the content of a natural image and thus, it creates artistic outputs. The 'Trajectory Visualizer' tool was inspired by the Gestalt law of continuity and visualizes the trajectory joints of a motion image source. The MotiVo editor also provides 'Image Filters' and 'Comic Annotator' tools, appropriate for the creation of artistic contents via colour editing and element annotations. Finally, the 'Comic

Synthesizer' tool can synthesize all the edited outputs from the aforementioned tools and create a completed use casework of the available MotiVo's editor functionality.

The novelty behind the HMAV approach lies in the simplicity and ease of use as it addresses users with no relevant experience on image editing software and requires minimum expertise and knowledge from the user side. Moreover, the MotiVo editor exploits various motion visualization techniques and creates hybrid visualization and artistic outputs. Finally, as the MotiVo editor was built on a dockable UI library, the system is highly extensible and familiar to novice users.

Based on the experience gained on developing the MotiVo editor and the experiments performed in the context of implementing its visualization tools, it can be safely concluded that artistic visualization of human motion in 2D is technically feasible and can produce aesthetically pleasing results. Of course, human intervention is needed to orchestrate the appropriate selection of tools. In addition, through the gained experience of developing the MotiVo tools and the produced use cases, this Thesis provides a set of best practice guidelines to get the most out of these tools.

Planned future work includes tackling all unaddressed issues discovered during the evaluation cycles as well as the findings of the final heuristic evaluation. We are going to enhance the HMAV approach by exploring new advances in style transfer algorithms to provide even better visualizations. Furthermore, the existing tools are planned to be upgraded to produce even better visualizations.

Regarding the 'Motion Blender' tool, we intent to enhance the blending algorithms and create some pre-sets to optimize its usage in accordance to the guidelines. The 'Trajectory Visualizer' tool is going to be upgraded through the integration of richer drawing formats (i.e. colours, shapes) that will allow the system to draw more artistically pleasant trajectories as well as the users would be able to define on run-time the trajectory on the input image (i.e. moving points on the trajectory path). The 'Foreground Amplifier' tool due to its large scale functionality, induces many changes such as; better precision on the rectangle and mask cut functionalities, more available background options and the 'Eraser' option should be enabled for all the available backgrounds (not only for black and white) and provide many diameter sizes for precise control of erasing.

For the 'Comic Annotator' tool, we intent to extend the libraries of annotation elements to cover more use cases and application domains and users be able to create custom element collections. Moreover, for the comic/bubble texts, different font styles and sizes would be supported. The

‘Image Simplifier’ tool would offer pre-set options to users (i.e. Min effect, Med effect, Max effect depending on the preferred simplification intensity (i.e., “Low”, “Medium” and “High”). For the ‘Comic Synthesizer’ tool, we are planning to design various ready-to-use templates for alternative purposes and domains such as; movie posters, company or museum flyers etc. In addition, in the future users would be able to create custom user grid templates and save them in their own custom library.

Furthermore, a beneficial addition to the whole system could be the integration of 3D information to the static motion frames as it will enable the use of 3D data, in favour of motion analytics and visualization. Finally, another extension of this work could be the creation of video visualizations such as summarizations and even the isolation of the actors in a movement sequence would be facilitated, to reproduce them in another context for example in VR training.

Bibliography

[1] S. Zeki, “Inner vision: An exploration of art and the brain,” 2002.

[2] R. Blake and M. Shiffrar, “Perception of human motion,” *Annu. Rev. Psychol.*, vol. 58, 2007.

[3] B. Pinna and A. Reeves, “From perception to art: How vision creates meanings,” *Spat. Vis.*, vol. 22, no. 3, pp. 225–272, 2009.

[4] G. Bush, P. Luu, and M. I. Posner, “Cognitive and emotional influences in anterior cingulate cortex,” *Trends Cogn. Sci.*, vol. 4, no. 6, pp. 215–222, 2000.

[5] L. Itti and C. Koch, “Computational modelling of visual attention,” *Nat. Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, 2001.

[6] A. M. Treisman and G. Gelade, “A feature-integration theory of attention,” *Cognit. Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.

[7] W. Köhler, “Gestalt psychology,” *Psychol. Forsch.*, vol. 31, no. 1, p. XVIII–XXX, 1967.

[8] F. Marini and C. A. Marzi, “Gestalt perceptual organization of visual stimuli captures attention automatically: Electrophysiological evidence,” *Front. Hum. Neurosci.*, vol. 10, p. 446, 2016.

[9] G. Johansson, “Visual motion perception,” *Sci. Am.*, vol. 232, no. 6, pp. 76–89, 1975.

[10] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, and M. Singh, *A Century of Gestalt Psychology in Visual Perception I. Perceptual Grouping and Figure-Ground Organization*.

[11] R. Kimchi and M. A. Peterson, “Figure-Ground Segmentation Can Occur Without Attention,” vol. 19, no. 7, p. 16.

[12] A. Bowler, “Politics as art: Italian Futurism and Fascism,” *Theory Soc.*, vol. 20, no. 6, pp. 763–794, Dec. 1991, doi: 10.1007/BF00678096.

[13] G.-D. Chen, C.-W. Lin, and H.-W. Fan, “The history and evolution of kinetic art,” *Int. J. Soc. Sci. Humanity*, vol. 5, no. 11, p. 922, 2015.

[14] E. Segel and J. Heer, “Narrative visualization: Telling stories with data,” *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1139–1148, 2010.

[15] S. McCloud, “Understanding comics: The invisible art,” *Northamp. Mass*, 1993.

[16] J. E. Cutting, “Representing Motion in a Static Image: Constraints and Parallels in Art, Science, and Popular Culture,” *Perception*, vol. 31, no. 10, pp. 1165–1193, Oct. 2002, doi: 10.1068/p3318.

[17] M. P. lndranil Chakravarty, “Modeling Motion Blur in Computer-Generated Images,” Jul. 1983.

[18] G. J. Brostow and I. Essa, “Image-based motion blur for stop motion animation,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 561–566.

[19] X. Zabulis *et al.*, “Representation and Preservation of Heritage Crafts,” *Sustainability*, vol. 12, no. 4, p. 1461, 2020.

[20] X. Zabulis *et al.*, “What is needed to digitise knowledge on Heritage Crafts?,” *Memoriamedia Rev.*, 2019.

[21] E. Panofsky, A. Dürer, and G. Vasari, *Meaning in the visual arts: papers in and on art history*, vol. 39. Doubleday Garden City, NY, 1955.

[22] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, “Documentation mocap database hdm05,” 2007.

[23] P. H. Lindsay and D. A. Norman, *Human information processing: An introduction to psychology*. Academic press, 2013.

[24] J. L. Brooks, “Traditional and new principles of perceptual grouping,” 2015.

[25] E. Rubin, *Visuell wahrgenommene figuren*. Рипол Классик, 1980.

[26] D. Zavagno and O. Daneyko, “Perceptual Grouping, and Color,” in *Encyclopedia of Color Science and Technology*, R. Luo, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1–5. doi: 10.1007/978-3-642-27851-8_220-1.

[27] M. H. Herzog, “Perceptual grouping,” *Curr. Biol.*, vol. 28, no. 12, pp. R687–R688, Jun. 2018, doi: 10.1016/j.cub.2018.04.041.

[28] S. Grossberg and E. M. Golla, “Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentations,” p. 31.

[29] M. B. Ben-Av, D. Sagi, and J. Braun, “Visual attention and perceptual grouping,” *Percept. Psychophys.*, vol. 52, no. 3, pp. 277–294, May 1992, doi: 10.3758/BF03209145.

[30] C. J. Darwin, “Auditory grouping,” *Trends Cogn. Sci.*, vol. 1, no. 9, pp. 327–333, Dec. 1997, doi: 10.1016/S1364-6613(97)01097-8.

[31] M. A. Schweisfurth, R. Schweizer, and S. Treue, “Feature-based attentional modulation of orientation perception in somatosensation,” *Front. Hum. Neurosci.*, vol. 8, 2014, doi: 10.3389/fnhum.2014.00519.

[32] F. L. Dimmick, “An Experimental Study of Visual Movement and the Phi Phenomenon,” *Am. J. Psychol.*, vol. 31, no. 4, pp. 317–332, 1920, doi: 10.2307/1413667.

[33] S. M. Axstis, “PHI MOVEMENT AS A SUBTRACTION PROCESS,” p. 21.

[34] M. Wertheimer, “Untersuchungen zur Lehre von der Gestalt. II,” *Psychol. Forsch.*, vol. 4, no. 1, pp. 301–350, 1923, doi: 10.1007/BF00410640.

[35] M. Kubovy and M. Van Den Berg, “The whole is equal to the sum of its parts: A probabilistic model of grouping by proximity and similarity in regular patterns.,” *Psychol. Rev.*, vol. 115, no. 1, p. 131, 2008.

[36] S. Han and G. W. Humphreys, “Interactions between perceptual organization based on Gestalt laws and those based on hierarchical processing,” *Percept. Psychophys.*, vol. 61, no. 7, pp. 1287–1298, Jan. 1999, doi: 10.3758/BF03206180.

[37] T. Tversky, W. S. Geisler, and J. S. Perry, “Contour grouping: Closure effects are explained by good continuation and proximity,” *Vision Res.*, vol. 44, no. 24, pp. 2769–2777, 2004.

[38] B. Spehar, “The role of contrast polarity in perceptual closure,” *Vision Res.*, vol. 42, no. 3, pp. 343–350, 2002.

[39] R. Hess, A. Hayes, and D. Field, “The roles of polarity and symmetry in the perceptual grouping of contour fragments,” *Spat. Vis.*, vol. 13, no. 1, pp. 51–66, 2000.

[40] A. B. Sekuler and P. J. Bennett, “Generalized common fate: Grouping by common luminance changes,” *Psychol. Sci.*, vol. 12, no. 6, pp. 437–444, 2001.

[41] S. P. Vecera and R. C. O’reilly, “Figure-ground organization and object recognition processes: an interactive account.,” *J. Exp. Psychol. Hum. Percept. Perform.*, vol. 24, no. 2, p. 441, 1998.

[42] P. J. Kellman and T. F. Shipley, "A theory of visual interpolation in object perception," *Cognit. Psychol.*, vol. 23, no. 2, pp. 141–221, 1991.

[43] C. C. Fowlkes, D. R. Martin, and J. Malik, "Local figure–ground cues are valid for natural images," *J. Vis.*, vol. 7, no. 8, pp. 2–2, 2007.

[44] X. Ren and J. Malik, "Tracking as Repeated Figure/Ground Segmentation," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8. doi: 10.1109/CVPR.2007.383177.

[45] W. Zou, C. Bai, K. Kpalma, and J. Ronsin, "Online Glocal Transfer for Automatic Figure-Ground Segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2109–2121, May 2014, doi: 10.1109/TIP.2014.2312287.

[46] S. P. Ambulkar and N. S. Sakhare, "Figure-Ground Segmentation Techniques," vol. 4, no. 11, p. 5, 2013.

[47] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 46–46.

[48] E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," in *European conference on computer vision*, 2002, pp. 109–122.

[49] S. L. Friedman and M. B. Stevenson, "Perception of movement in pictures," *Percept. Pict.*, vol. 1, pp. 225–255, 1980.

[50] D. Weiskopf, "On the role of color in the perception of motion in animated visualizations," in *IEEE Visualization 2004*, 2004, pp. 305–312.

[51] E. H. Gombrich, *Art and illusion*. Pantheon Books New York, 1961.

[52] H. Kim and G. Francis, "A computational and perceptual account of motion lines," *Perception*, vol. 27, no. 7, pp. 785–797, 1998.

[53] D. Burr, "Motion vision: Are 'speed lines' used in human visual motion?," *Curr. Biol.*, vol. 10, no. 12, pp. R440–R443, 2000.

[54] J. E. Cutting, "Images, Imagination, and Movement: Pictorial Representations and Their Development in the Work of James Gibson," *Perception*, vol. 29, no. 6, pp. 635–648, Jun. 2000, doi: 10.1068/p2976.

[55] N. Cohn and S. Maher, "The notion of the motion: The neurocognition of motion lines in visual narratives," *Brain Res.*, vol. 1601, pp. 73–84, 2015.

[56] F. Navarro, F. J. Serón, and D. Gutierrez, "Motion Blur Rendering: State of the Art," *Comput. Graph. Forum*, vol. 30, no. 1, pp. 3–26, Mar. 2011, doi: 10.1111/j.1467-8659.2010.01840.x.

[57] E. Muybridge, *The human figure in motion*. Courier Corporation, 2012.

[58] S. McCloud, *Making comics: Storytelling secrets of comics, manga and graphic novels*. Harper New York, 2006.

[59] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 125–141, 2008.

[60] J. Assa, Y. Caspi, and D. Cohen-Or, "Action synopsis: pose selection and illustration," in *ACM Transactions on Graphics (TOG)*, 2005, vol. 24, pp. 667–676.

[61] M. Wang, S. Guo, M. Liao, D. He, J. Chang, and J. Zhang, "Action snapshot with single pose and viewpoint," *Vis. Comput.*, vol. 35, no. 4, pp. 507–520, 2019.

[62] H. Yasuda, R. Kaihara, S. Saito, and M. Nakajima, "Motion belts: Visualization of human motion data on a timeline," *IEICE Trans. Inf. Syst.*, vol. 91, no. 4, pp. 1159–1167, 2008.

[63] S. Bouvier-Zappa, V. Ostromoukhov, and P. Poulin, "Motion cues for illustration of skeletal motion capture data," in *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, 2007, pp. 133–140.

[64] O. Teramoto, I. K. Park, and T. Igarashi, "Interactive motion photography from a single image," *Vis. Comput.*, vol. 26, no. 11, pp. 1339–1348, 2010.

[65] M. Masuch, S. Schlechtweg, and R. Schulz, "R.: Speedlines: Depicting motion in motionless pictures," 1999.

[66] M. G. Choi, K. Yang, T. Igarashi, J. Mitani, and J. Lee, "Retrieval and visualization of human motion data via stick figures," in *Computer Graphics Forum*, 2012, vol. 31, pp. 2057–2065.

[67] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande, "Turning to the masters: motion capturing cartoons," *ACM Trans. Graph. TOG*, vol. 21, no. 3, pp. 399–407, 2002.

[68] Y. Kawagishi, K. Hatsuyama, and K. Kondo, "Cartoon blur: nonphotorealistic motion blur," in *Proceedings Computer Graphics International 2003*, 2003, pp. 276–281.

[69] M. Nienhaus and J. Dollner, "Depicting dynamics using principles of visual art and narrations," *IEEE Comput. Graph. Appl.*, vol. 25, no. 3, pp. 40–51, 2005.

[70] W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Motion without movement," *ACM Siggraph Comput. Graph.*, vol. 25, no. 4, pp. 27–30, 1991.

[71] M.-T. Chi, T.-Y. Lee, Y. Qu, and T.-T. Wong, "Self-animating images: illusory motion using repeated asymmetric patterns," *ACM Trans. Graph. TOG*, vol. 27, no. 3, pp. 1–8, 2008.

[72] M. Lockyer and L. Bartram, "The aMotion Toolkit: Painting with affective motion textures," in *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, 2012, pp. 35–43.

[73] B. Kim and I. Essa, "Video-based nonphotorealistic and expressive illustration of motion," in *International 2005 Computer Graphics*, 2005, pp. 32–35.

[74] J. P. Collomosse, D. Rowntree, and P. M. Hall, "Rendering cartoon-style motion cues in post-production video," *Graph. Models*, vol. 67, no. 6, pp. 549–564, 2005.

[75] D. Kim, B. Chakravarthi, S. H. Kim, A. Balasubramanyam, Y. H. Chai, and A. K. Patil, "MotionNote: A Novel Human Pose Representation," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2020, pp. 697–698.

[76] K.-S. Huang, C.-F. Chang, Y.-Y. Hsu, and S.-N. Yang, "Key probe: a technique for animation keyframe extraction," *Vis. Comput.*, vol. 21, no. 8–10, pp. 532–541, 2005.

[77] R. H. Kazi, T. Grossman, C. Mogk, R. Schmidt, and G. Fitzmaurice, "ChronoFab: Fabricating Motion," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA, May 2016, pp. 908–918. doi: 10.1145/2858036.2858138.

[78] A. Lake, C. Marshall, M. Harris, and M. Blackstein, "Stylized rendering techniques for scalable real-time 3d animation," in *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, 2000, pp. 13–20.

[79] H. Yang and K. Min, "Importance-based approach for rough drawings," *Vis. Comput.*, vol. 35, no. 4, pp. 609–622, 2019.

[80] M. Haller, C. Hanl, and J. Diephuis, "Non-photorealistic rendering techniques for motion in computer games," *Comput. Entertain. CIE*, vol. 2, no. 4, pp. 11–11, 2004.

[81] X. Zhang *et al.*, "Mosculp: Interactive visualization of shape and time," in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 2018, pp. 275–285.

[82] P.-Y. (Peggy) Chi, D. Vogel, M. Dontcheva, W. Li, and B. Hartmann, "Authoring Illustrations of Human Movements by Iterative Physical Demonstration," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, Tokyo Japan, Oct. 2016, pp. 809–820. doi: 10.1145/2984511.2984559.

[83] E. Stefanidi, N. Partarakis, X. Zabulis, P. Zikas, G. Papagiannakis, and N. Thalmann Magnenat, "TooltY: An approach for the combination of motion capture and 3D reconstruction to present tool usage in 3D environments," in *Intelligent Scene Modelling and Human Computer Interaction*, Springer.

[84] M. Thorne, D. Burke, and M. van de Panne, "Motion Doodles: An Interface for Sketching Character Motion," p. 8.

[85] R. W. Sumner, H. Bowles, and M. Gross, "Programmable Motion Effects," p. 9.

[86] J. Chmelik and J. Sochor, "Body motion visualization in virtual environment," in *Proceedings of the Conference on Geometry and Graphics*, 2010, pp. 119–124.

[87] D. Falie, M. Ichim, and L. David, "Respiratory motion visualization and the sleep apnea diagnosis with the time of flight (ToF) camera," *Vis. Imaging Simul.*, 2008.

[88] K. Xu *et al.*, "Development of a 3D tongue motion visualization platform based on ultrasound image sequences," *ArXiv Prepr. ArXiv160506106*, 2016.

[89] S.-M. Choi and M.-H. Kim, “Motion visualization of human left ventricle with a time-varying deformable model for cardiac diagnosis,” *J. Vis. Comput. Animat.*, vol. 12, no. 2, pp. 55–66, 2001.

[90] H.-J. Lee, H. J. Shin, and J.-J. Choi, “Single image summarization of 3D animation using depth images,” *Comput. Animat. Virtual Worlds*, vol. 23, no. 3–4, pp. 417–424, 2012.

[91] Y. Shen, H. Wang, E. S. Ho, L. Yang, and H. P. Shum, “Posture-based and action-based graphs for boxing skill visualization,” *Comput. Graph.*, vol. 69, pp. 104–115, 2017.

[92] G. Pingali, A. Opalach, Y. Jean, and I. Carlbom, “Visualization of sports using motion trajectories: providing insights into performance, style, and strategy,” in *Proceedings Visualization, 2001. VIS’01.*, 2001, pp. 75–544.

[93] C. Kirmizibayrak, J. Honorio, X. Jiang, R. Mark, and J. K. Hahn, *Digital Analysis and Visualization of Swimming Motion*.

[94] J.-Y. Mao, K. Vredenburg, P. W. Smith, and T. Carey, “The state of user-centered design practice,” *Commun. ACM*, vol. 48, no. 3, pp. 105–109, 2005.

[95] C. Abras, D. Maloney-Krichmar, and J. Preece, “User-centered design,” *Bainbridge W Encycl. Hum.-Comput. Interact. Thousand Oaks Sage Publ.*, vol. 37, no. 4, pp. 445–456, 2004.

[96] “GitHub - DevZest/WpfDocking: A docking library to integrate undo/redo-able tabbed docking, floating and auto hide window management into your application in minutes.” https://github.com/DevZest/WpfDocking (accessed Jan. 13, 2020).

[97] Anastasia Rigaki, Nikolaos Partarakis, Xenophon Zabulis, and Constantine Stephanidis, “An Approach Towards Artistic Visualizations of Human Motion in Static Media Inspired by the Visual Arts,” presented at the ACHI 2020 : The Thirteenth International Conference on Advances in Computer-Human Interactions, Valencia, Spain.

[98] C. Rother, V. Kolmogorov, and A. Blake, “‘GrabCut’ — Interactive Foreground Extraction using Iterated Graph Cuts,” p. 6.

[99] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, Inc., 2008.

[100] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vancouver, BC, Canada, 2001, vol. 1, pp. 105–112. doi: 10.1109/ICCV.2001.937505.

[101] Q. Zhang, X. Shen, L. Xu, and J. Jia, “Rolling guidance filter,” in *European conference on computer vision*, 2014, pp. 815–830.

[102] Q. Zhang, X. Shen, L. Xu, and J. Jia, “Rolling Guidance Filter,” in *Computer Vision – ECCV 2014*, vol. 8691, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 815–830. doi: 10.1007/978-3-319-10578-9_53.

[103] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, 1998, pp. 839–846.

[104] E. V. Shikin and A. I. Plis, *Handbook on Splines for the User*. CRC Press, 1995.

[105] J. Nielsen, “Iterative user-interface design,” *Computer*, vol. 26, no. 11, pp. 32–41, 1993.

[106] J. Rieman, M. Franzke, and D. Redmiles, “Usability evaluation with the cognitive walkthrough,” in *Conference companion on Human factors in computing systems*, 1995, pp. 387–388.

[107] J. Nielsen and R. Molich, “Heuristic evaluation of user interfaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1990, pp. 249–256.

[108] J. Nielsen, “Finding usability problems through heuristic evaluation,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 373–380.

Appendix A

Comic Book (Version 1): The making of a glass carafe

The comic book: This is an illustrated book that presents the completed test use of the MotiVo editor regarding the creation of the Bontemps carafe. The illustrated figures where created using the MotiVo editor's tools and by facilitating key-frames extracted by video recordings of the creation process at CERVAF. For the selection of key-frames the ethnographic research conducted by CNAM at CERVAF was used as a guide of identifying the major steps of the carafe making process and the important moves / gestures that should be highlighted in the comic book.

The making of the Bontemps' carafe

Contributors

Anastasia Rigaki (FORTH)

Comic rendering and motion visualization using the MotiVo editor

Arnaud Dubois (CNAM)

Ethnographic research, craft understanding and craft modelling, definition of motion driven narrative and video segmentation

Jean-Pierre Mateus (CERFAV)

Glassmaking demonstration

Ammar Qammaz (FORTH)

Video recording of craft demonstration

Brenda Olivias, Dimitrios Menychtas and Alina Glushkova (ARMINES)

Craft recording and MoCap

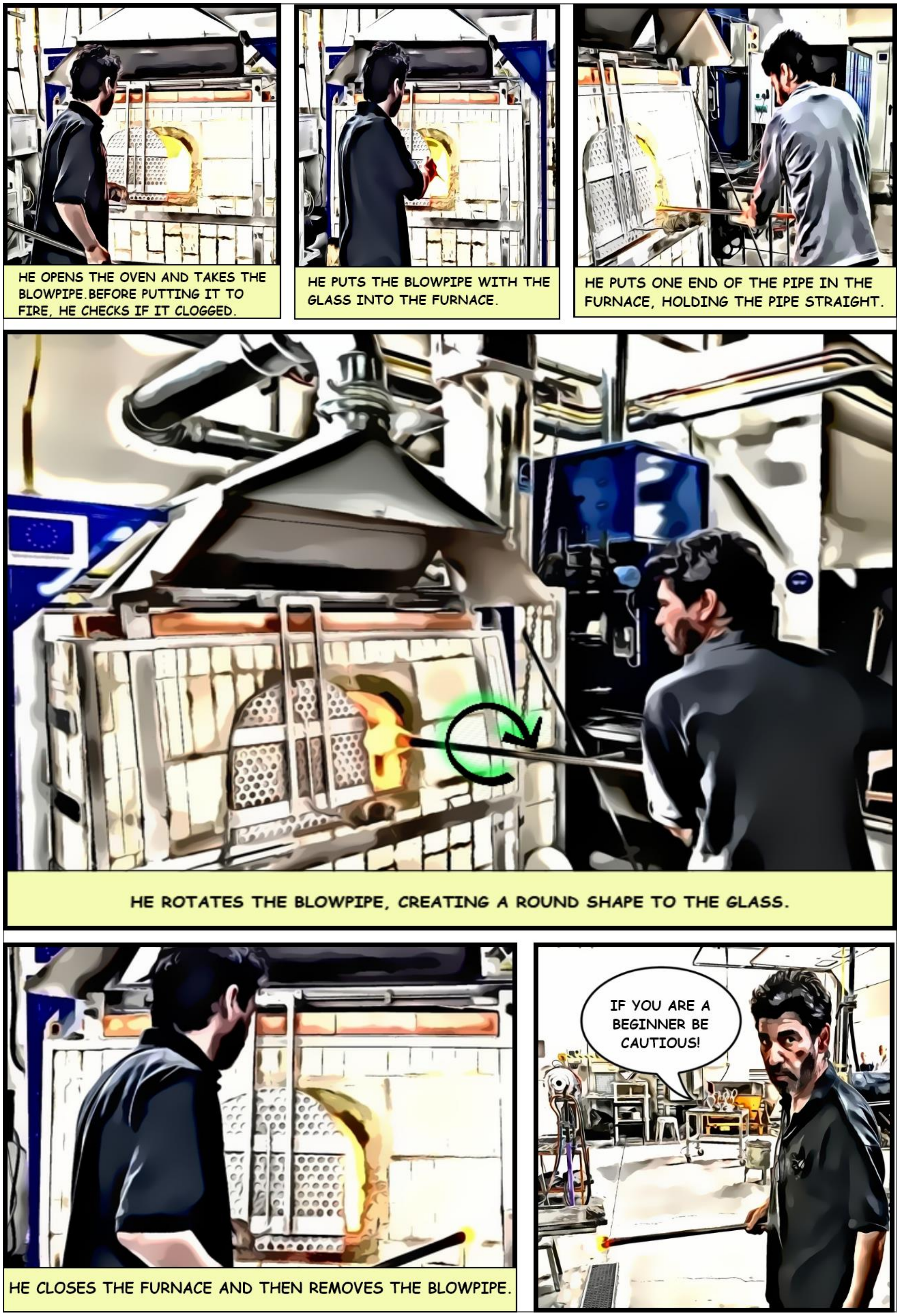
Xenophon Zabulis & Nikolaos Partarakis (FORTH)

Concept definition

The Glass Master

Jean-Pierre Mateus

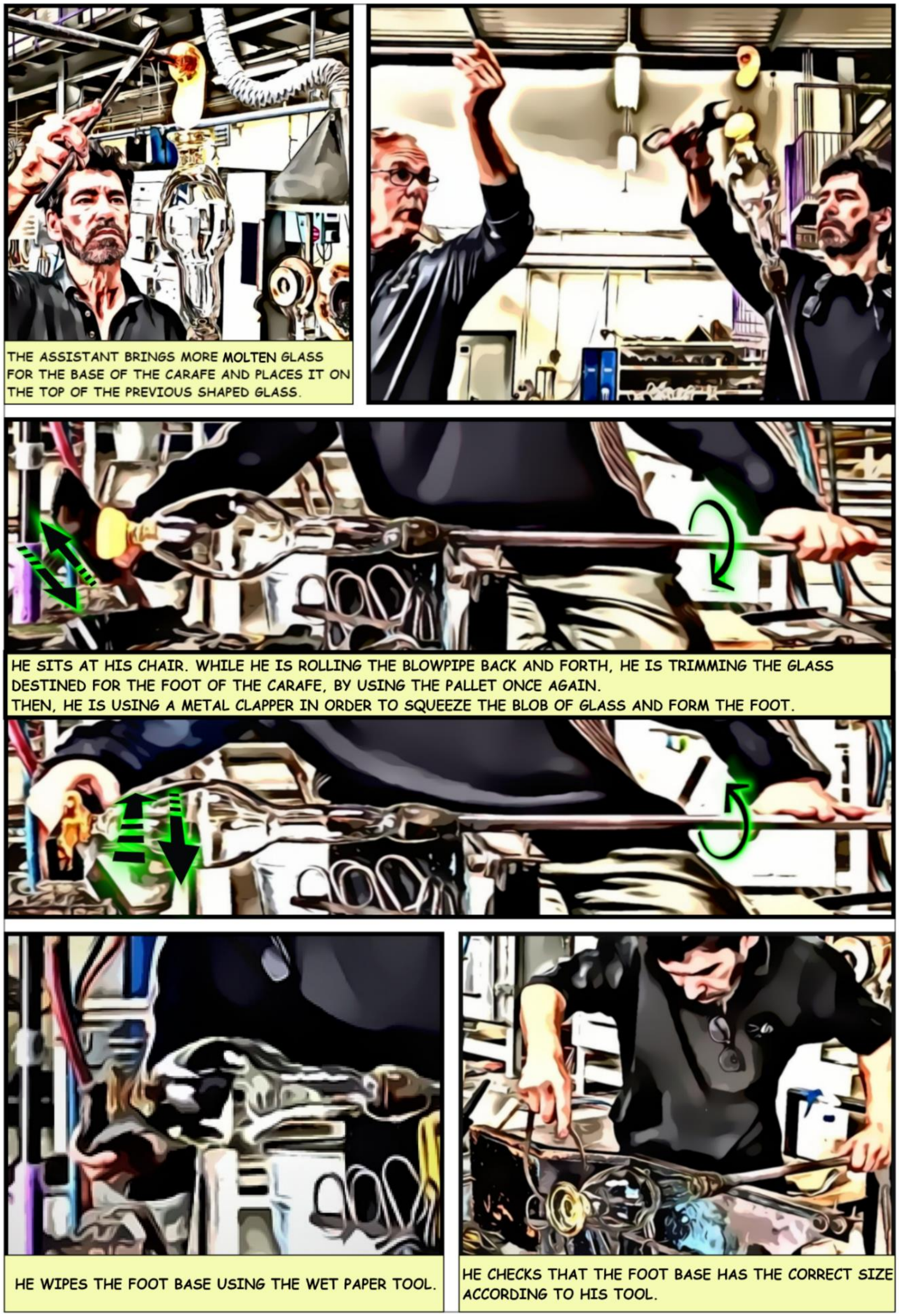
"We can't tame this craft"


HE OPENS THE OVEN AND TAKES THE BLOWPIPE.BEFORE PUTTING IT TO FIRE, HE CHECKS IF IT CLOGGED.
HE PUTS THE BLOWPIPE WITH THE GLASS INTO THE FURNACE.
HE PUTS ONE END OF THE PIPE IN THE FURNACE, HOLDING THE PIPE STRAIGHT.
HE ROTATES THE BLOWPIPE, CREATING A ROUND SHAPE TO THE GLASS.
HE CLOSES THE FURNACE AND THEN REMOVES THE BLOWPIPE.
IF YOU ARE A BEGINNER BE CAUTIOUS!

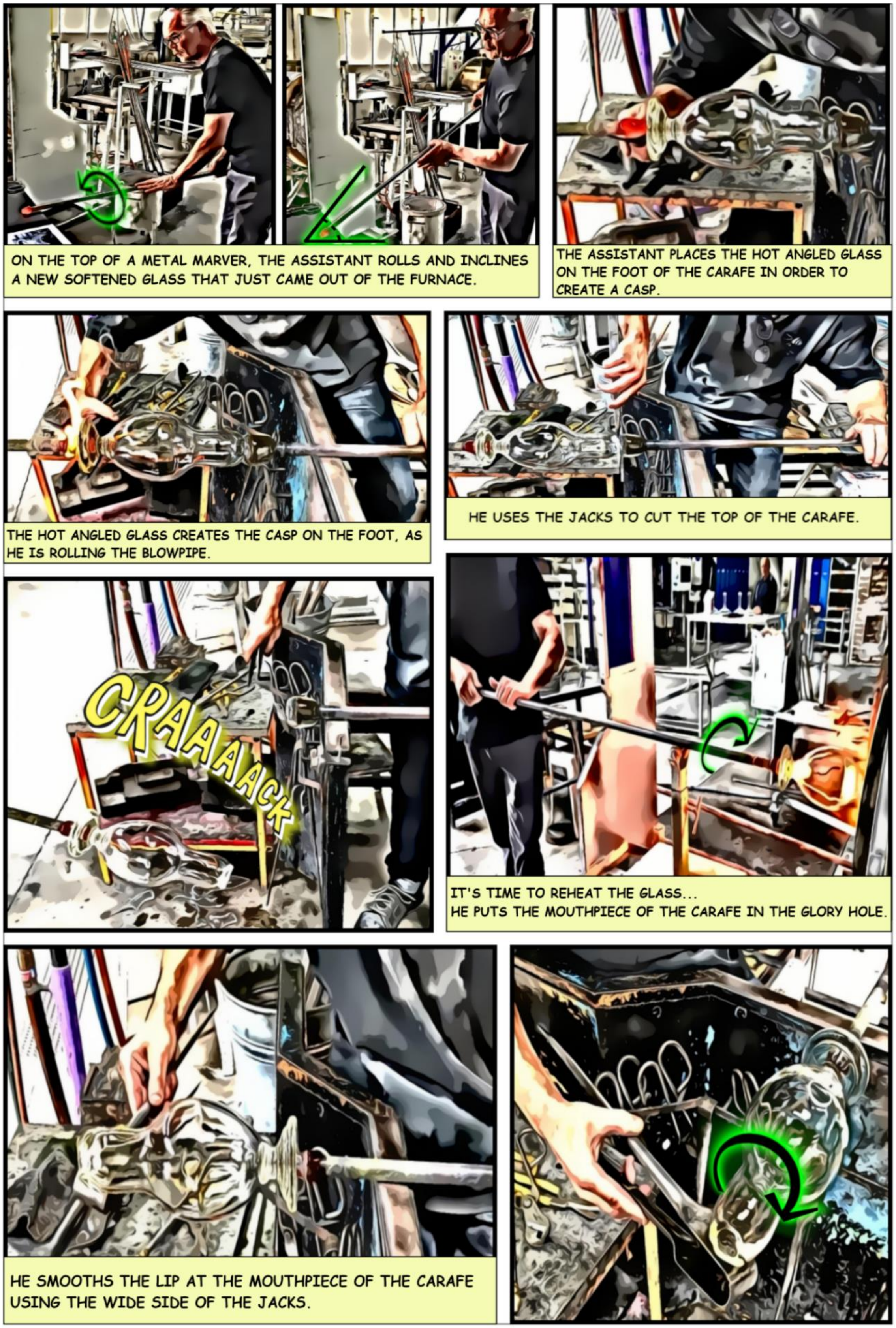

FIRST I HAVE TO COOL THE BLOWPIPE!
HE SITS AT HIS CHAIR AND WATERS HIS PAPER TOOL.
HE ROLLS THE BLOWPIPE WHILE HE IS WIPING THE GLASS WITH THE WET PAPER TOOL.
HE STANDS UP AND BLOWS INTO THE PIPE FOR 10 TO 15 SECONDS. AT THE SAME TIME HE ROLLS THE PIPE.
HE ROLLS BACK AND FORTH THE BLOWPIPE WHILE STANDING.
IT'S TIME FOR A SECOND TIME IN THE FURNACE.
AFTER THAT, HE ROLLS THE BLOWPIPE AND WHIPES THE GLASS WITH THE WET PAPER TOOL.
ONCE AGAIN, HE BLOWS THROUGH THE BLOWPIPE AND THEN HE STRECHES...
HE USES A METAL CARVED MOLD IN ORDER THE SHAPE AND TEXTURE OF THE GLASS TO MATCH ITS DESIGN.

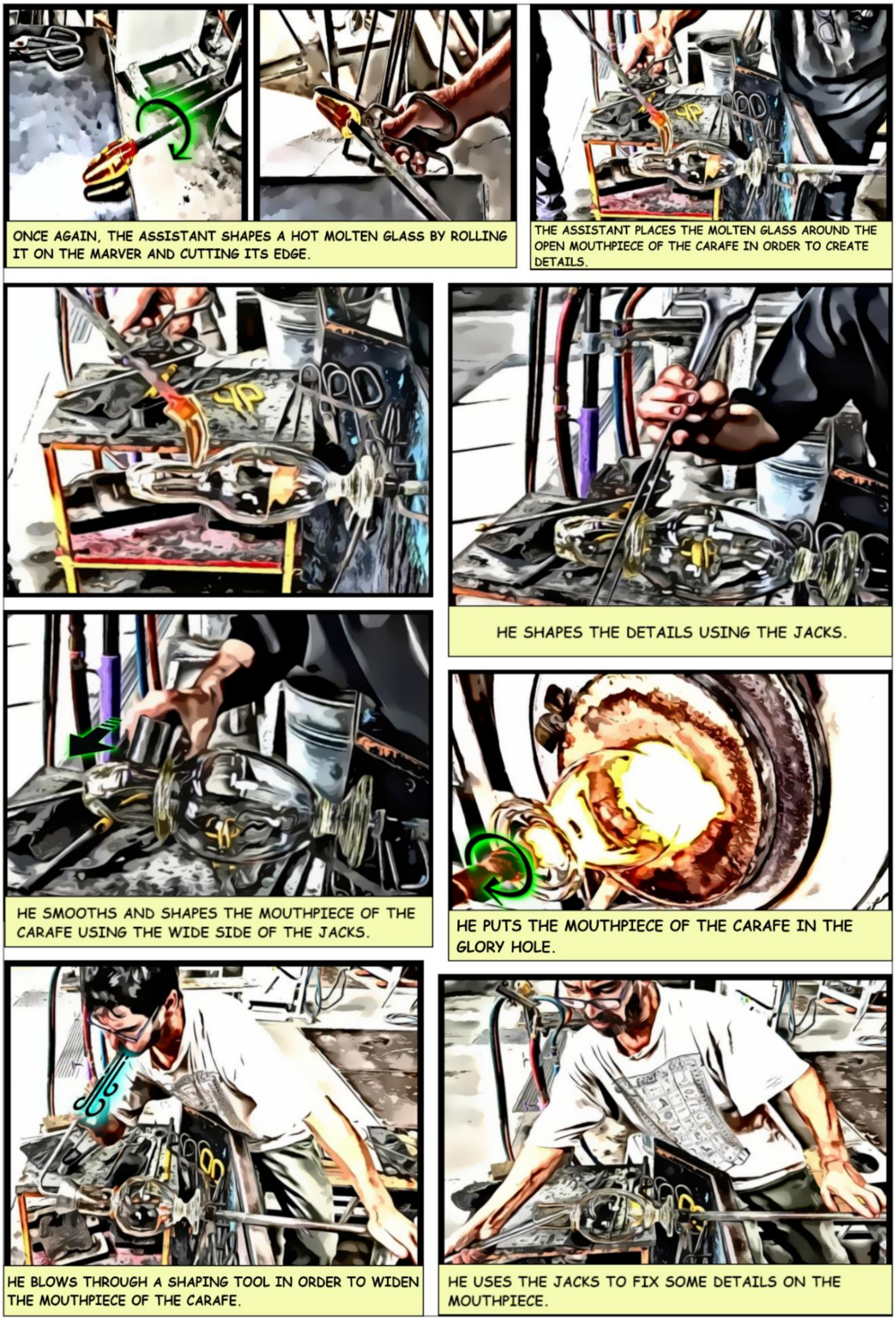

HE IS WIPING THE GLASS WITH THE WET PAPER AS HE IS ROLLING THE PIPE. AT THE SAME TIME, HIS ASSISTANT BLOWS THROUGH THE PIPE AND INFLATES THE BUBBLE OF GLASS.
HE USES METAL JACKS TO SHAPE THE GLASS.
AFTER RE-HEATING THE GLASS, HE BLOWS THROUGH THE PIPE FOR A COUPLE OF TIMES.
HE CHECKS IF THE GLASS HAS THE APPROPRIATE SIZE ACCORDING TO THE PATTERN SHAPE.
HE BALANCES AND WIPES THE GLASS WITH THE WET PAPER WHILE HE IS ROLLING THE BLOWPIPE.

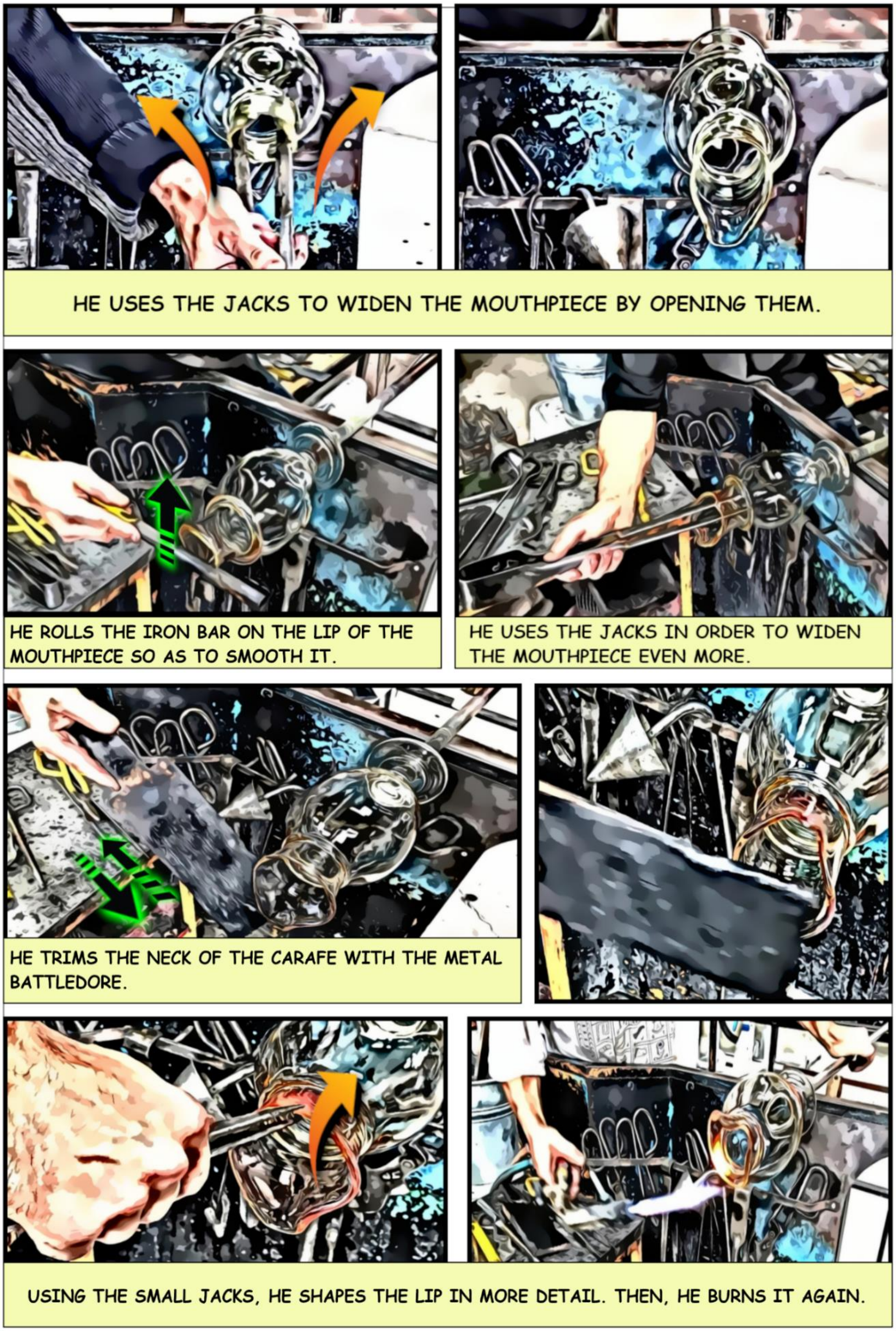

IT'S TIME FOR THE GLASS MASTER TO CREATE THE CARAFE BASE...
THE ASSISTANT TAKES OUT OF THE FURNACE A NEW MOLTEN BLOB OF GLASS AND HE IS CHECKING THE EXACT POSITION THAT SHOULD BE PLACED.
HE PLACES THE MOLTEN GLASS ON THE INFLATED GLASS AND CUTS IT WITH METAL SHEARS.
HE PUSHES THE MOLTEN GLASS ON THE INFLATED GLASS USING THE WIDE SIDE OF THE JACKS.
HE WIPES THE GLASS WITH THE WET PAPER. AFTER THAT, HE USES A METAL PALLET IN ORDER TO FLATTEN THE BASE OF THE CARAFE.



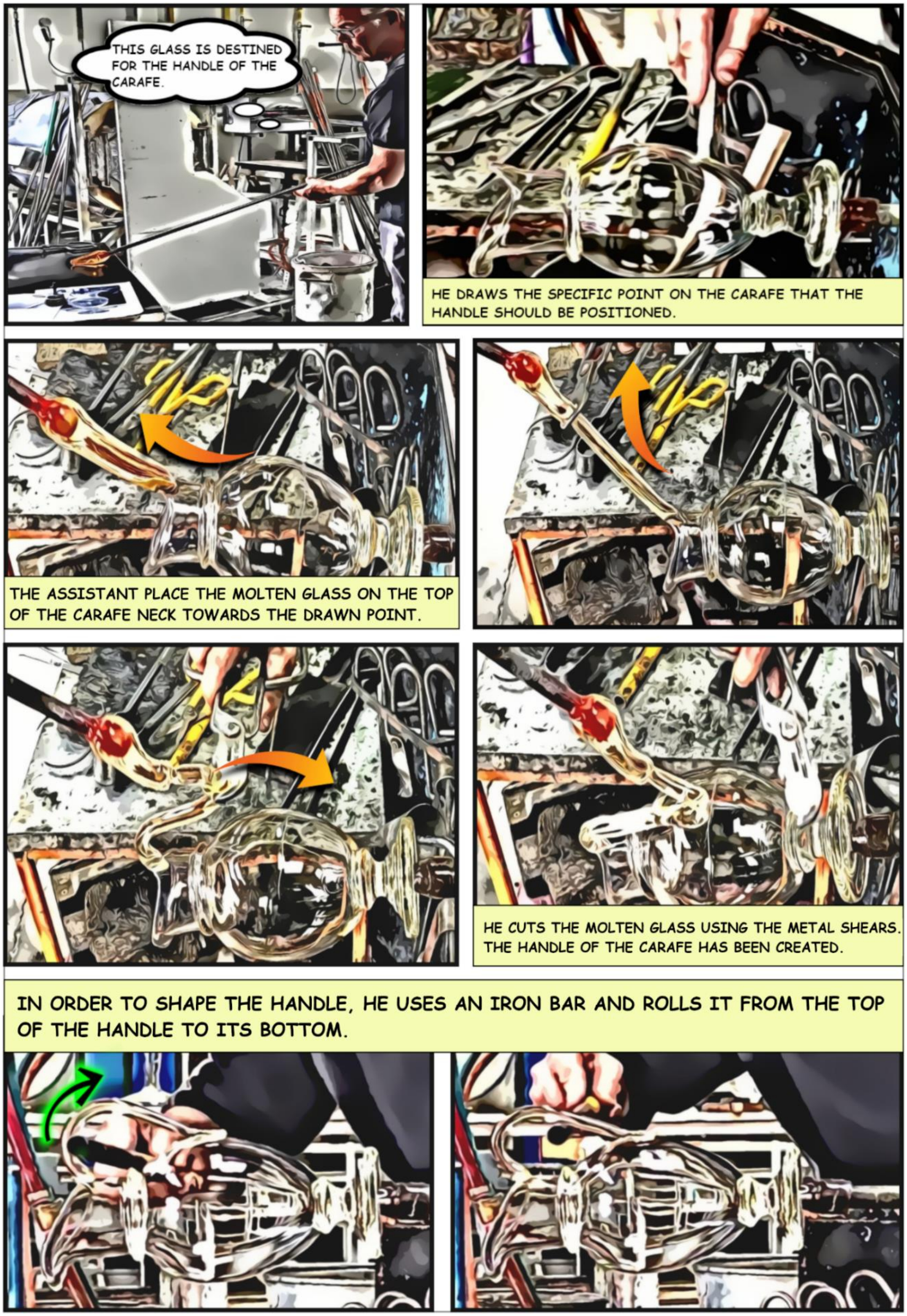
THE ASSISTANT BRINGS MORE MOLTEN GLASS FOR THE BASE OF THE CARAFE AND PLACES IT ON THE TOP OF THE PREVIOUS SHAPED GLASS.
HE SITS AT HIS CHAIR. WHILE HE IS ROLLING THE BLOWPIPE BACK AND FORTH, HE IS TRIMMING THE GLASS DESTINED FOR THE FOOT OF THE CARAFE, BY USING THE PALLET ONCE AGAIN.
THEN, HE IS USING A METAL CLAPPER IN ORDER TO SQUEEZE THE BLOB OF GLASS AND FORM THE FOOT.
HE WIPES THE FOOT BASE USING THE WET PAPER TOOL.
HE CHECKS THAT THE FOOT BASE HAS THE CORRECT SIZE ACCORDING TO HIS TOOL.

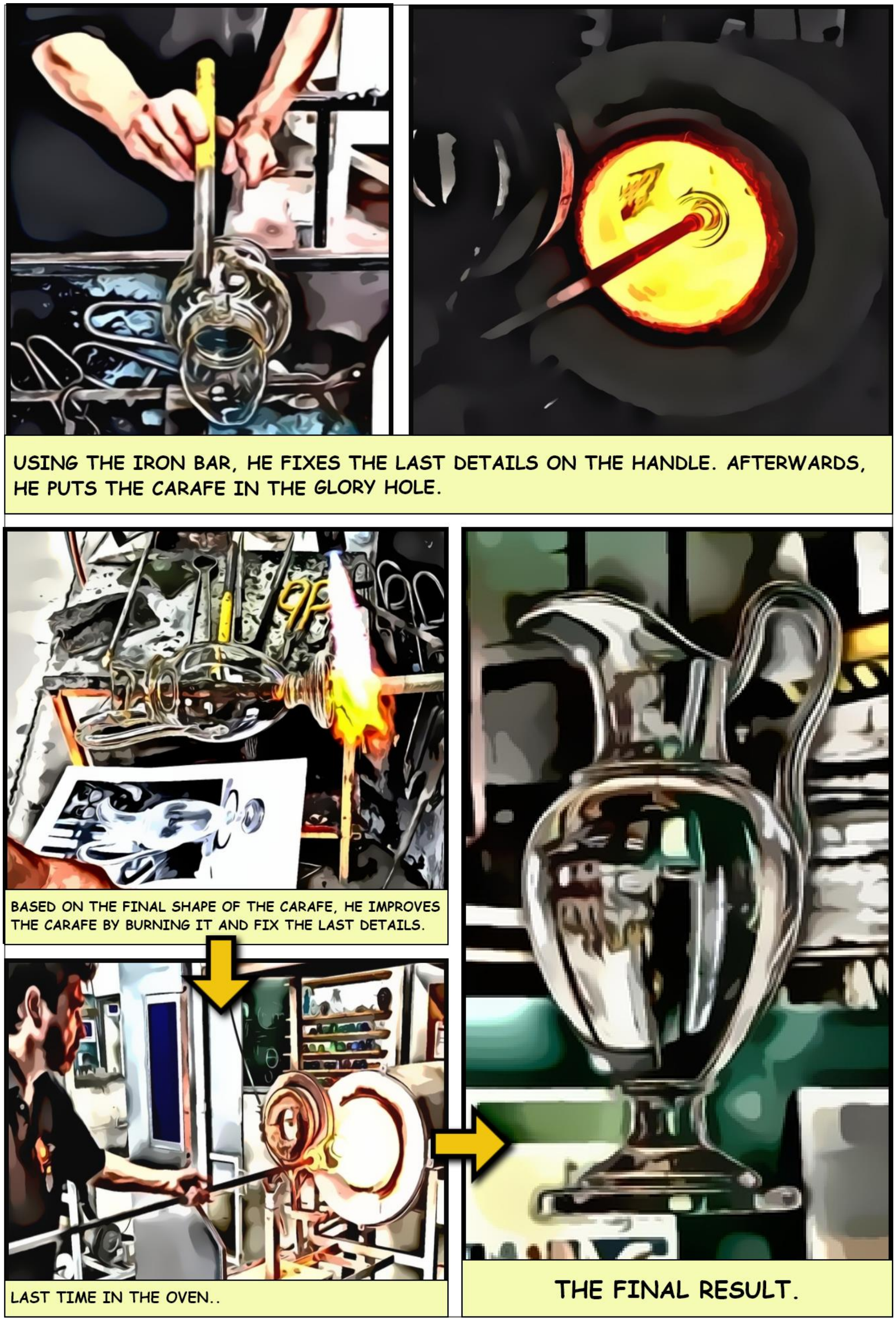

ON THE TOP OF A METAL MARVER, THE ASSISTANT ROLLS AND INCLINES A NEW SOFTENED GLASS THAT JUST CAME OUT OF THE FURNACE.
THE ASSISTANT PLACES THE HOT ANGLED GLASS ON THE FOOT OF THE CARAFE IN ORDER TO CREATE A CASP.
THE HOT ANGLED GLASS CREATES THE CASP ON THE FOOT, AS HE IS ROLLING THE BLOWPIPE.
HE USES THE JACKS TO CUT THE TOP OF THE CARAFE.
CRAAAACK
IT'S TIME TO REHEAT THE GLASS...
HE PUTS THE MOUTHPIECE OF THE CARAFE IN THE GLORY HOLE.
HE SMOOTHS THE LIP AT THE MOUTHPIECE OF THE CARAFE USING THE WIDE SIDE OF THE JACKS.


ONCE AGAIN, THE ASSISTANT SHAPES A HOT MOLTEN GLASS BY ROLLING IT ON THE MARVER AND CUTTING ITS EDGE.
THE ASSISTANT PLACES THE MOLTEN GLASS AROUND THE OPEN MOUTHPIECE OF THE CARAFE IN ORDER TO CREATE DETAILS.
HE SHAPES THE DETAILS USING THE JACKS.
HE SMOOTHS AND SHAPES THE MOUTHPIECE OF THE CARAFE USING THE WIDE SIDE OF THE JACKS.
HE PUTS THE MOUTHPIECE OF THE CARAFE IN THE GLORY HOLE.
HE BLOWS THROUGH A SHAPING TOOL IN ORDER TO WIDEN THE MOUTHPIECE OF THE CARAFE.
HE USES THE JACKS TO FIX SOME DETAILS ON THE MOUTHPIECE.


HE USES THE JACKS TO WIDEN THE MOUTHPIECE BY OPENING THEM.
HE ROLLS THE IRON BAR ON THE LIP OF THE MOUTHPIECE SO AS TO SMOOTH IT.
HE USES THE JACKS IN ORDER TO WIDEN THE MOUTHPIECE EVEN MORE.
HE TRIMS THE NECK OF THE CARAFE WITH THE METAL BATTLEDORE.
USING THE SMALL JACKS, HE SHAPES THE LIP IN MORE DETAIL. THEN, HE BURNS IT AGAIN.

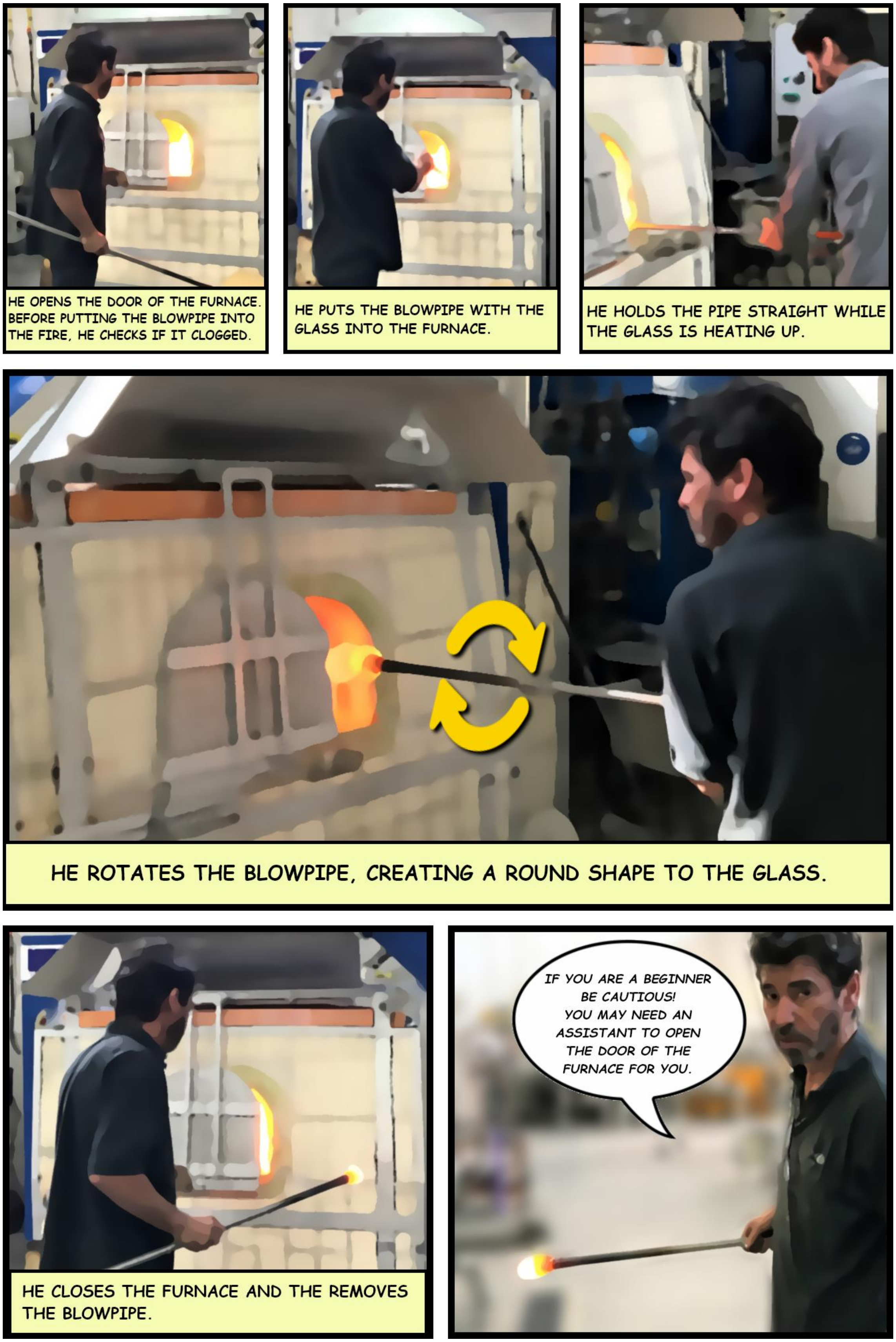

HE PUTS THE MOUTHPIECE IN THE GLORY HOLE FOR THE SECOND TIME.
THE LIP OF THE MOUTHPIECE HAS SOFTENED BY THE FIRE SO HE IS USING THE SMALL JACKS TO SHAPE IT.
HE CUTS THE EDGE OF THE LIP USING THE METAL SHEARS.
HE WARMS THE MOUTHPIECE IN ORDER TO MAKE IT SOFT AGAIN AND FURTHER SHAPE IT.
USING AN IRON BAR, HE SHAPES THE LIP OF THE MOUTHPIECE BY LIFTING IT UPWARDS.

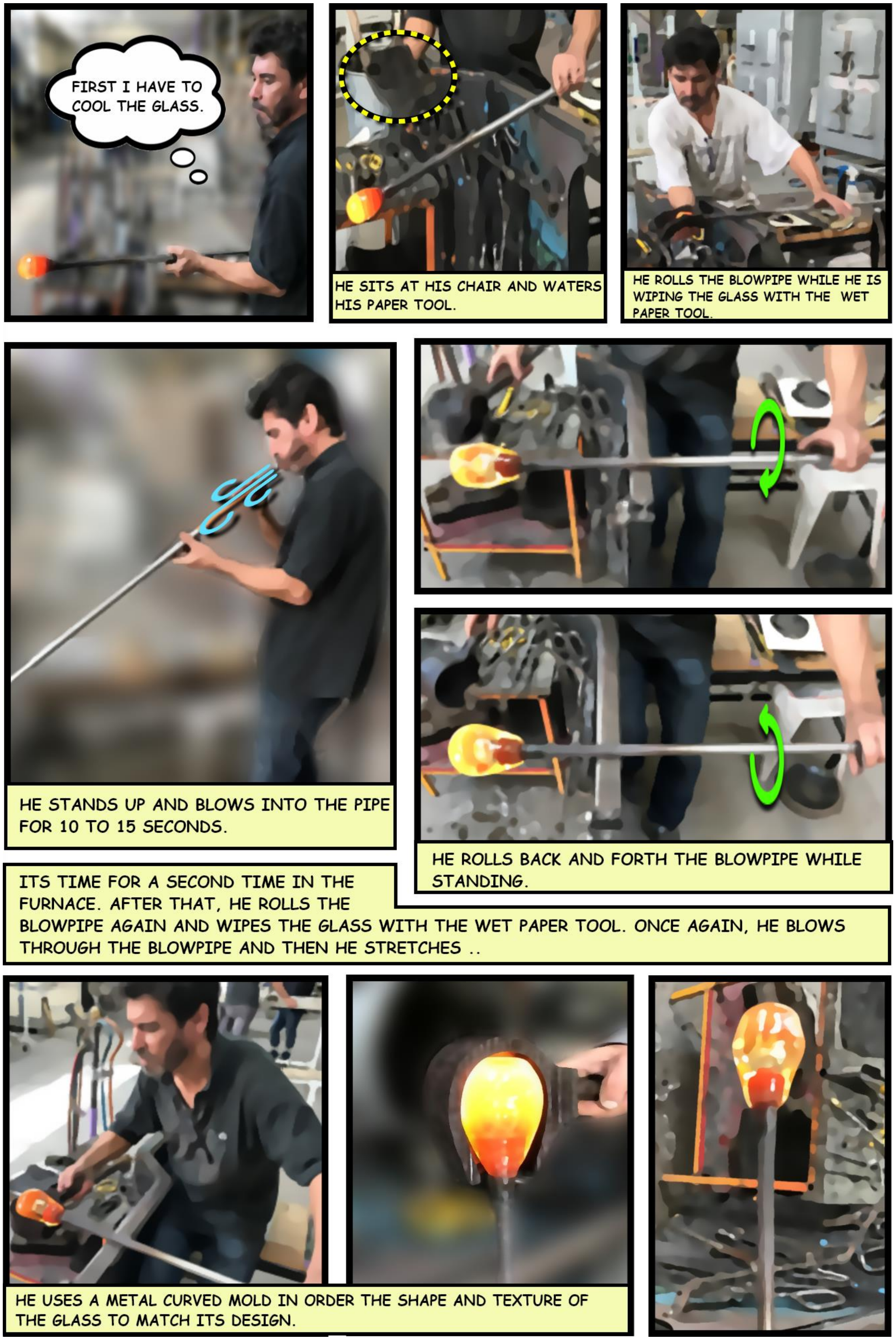

THIS GLASS IS DESTINED FOR THE HANDLE OF THE CARAFE.
HE DRAWS THE SPECIFIC POINT ON THE CARAFE THAT THE HANDLE SHOULD BE POSITIONED.
THE ASSISTANT PLACE THE MOLTEN GLASS ON THE TOP OF THE CARAFE NECK TOWARDS THE DRAWN POINT.
HE CUTS THE MOLTEN GLASS USING THE METAL SHEARS. THE HANDLE OF THE CARAFE HAS BEEN CREATED.
IN ORDER TO SHAPE THE HANDLE, HE USES AN IRON BAR AND ROLLS IT FROM THE TOP OF THE HANDLE TO ITS BOTTOM.

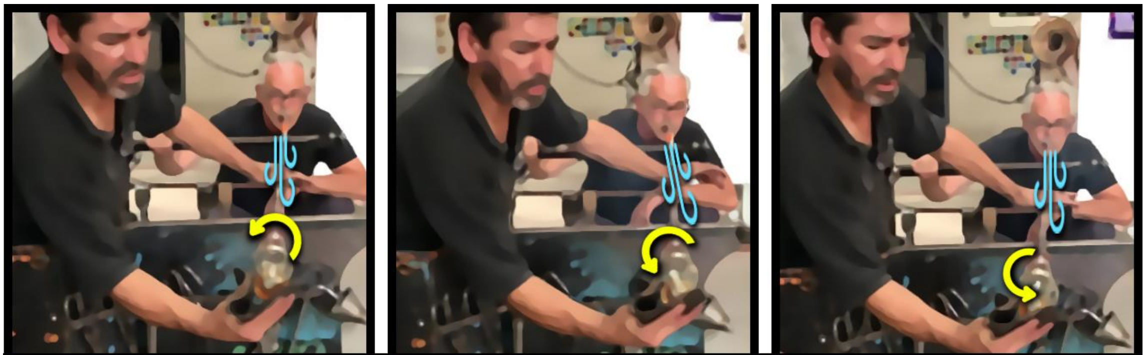

USING THE IRON BAR, HE FIXES THE LAST DETAILS ON THE HANDLE. AFTERWARDS, HE PUTS THE CARAFE IN THE GLORY HOLE.
BASED ON THE FINAL SHAPE OF THE CARAFE, HE IMPROVES THE CARAFE BY BURNING IT AND FIX THE LAST DETAILS.
LAST TIME IN THE OVEN..
THE FINAL RESULT.

Appendix B

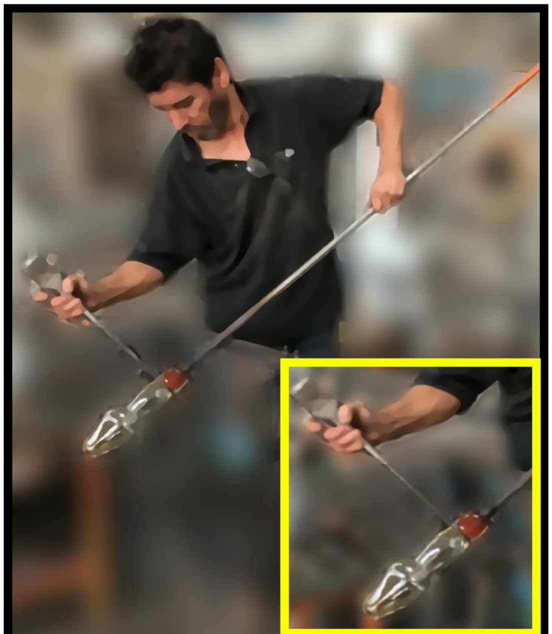
Comic Book (Version 2)


HE OPENS THE DOOR OF THE FURNACE. BEFORE PUTTING THE BLOWPIPE INTO THE FIRE, HE CHECKS IF IT CLOGGED.
HE PUTS THE BLOWPIPE WITH THE GLASS INTO THE FURNACE.
HE HOLDS THE PIPE STRAIGHT WHILE THE GLASS IS HEATING UP.
HE ROTATES THE BLOWPIPE, CREATING A ROUND SHAPE TO THE GLASS.
HE CLOSES THE FURNACE AND THE REMOVES THE BLOWPIPE.
IF YOU ARE A BEGINNER BE CAUTIOUS! YOU MAY NEED AN ASSISTANT TO OPEN THE DOOR OF THE FURNACE FOR YOU.


FIRST I HAVE TO COOL THE GLASS.
HE SITS AT HIS CHAIR AND WATERS HIS PAPER TOOL.
HE ROLLS THE BLOWPIPE WHILE HE IS WIPING THE GLASS WITH THE WET PAPER TOOL.
HE STANDS UP AND BLOWS INTO THE PIPE FOR 10 TO 15 SECONDS.
HE ROLLS BACK AND FORTH THE BLOWPIPE WHILE STANDING.
ITS TIME FOR A SECOND TIME IN THE FURNACE. AFTER THAT, HE ROLLS THE BLOWPIPE AGAIN AND WIPES THE GLASS WITH THE WET PAPER TOOL. ONCE AGAIN, HE BLOWS THROUGH THE BLOWPIPE AND THEN HE STRETCHES ..
HE USES A METAL CURVED MOLD IN ORDER THE SHAPE AND TEXTURE OF THE GLASS TO MATCH ITS DESIGN.



HE IS WIPING THE GLASS WITH THE WET PAPER AS HE IS ROLLING THE BLOWPIPE. AT THE SAME TIME, THE ASSISTANT BLOWS THROUGH THE BLOWPIPE AND INFLATES THE BUBBLE OF GLASS.

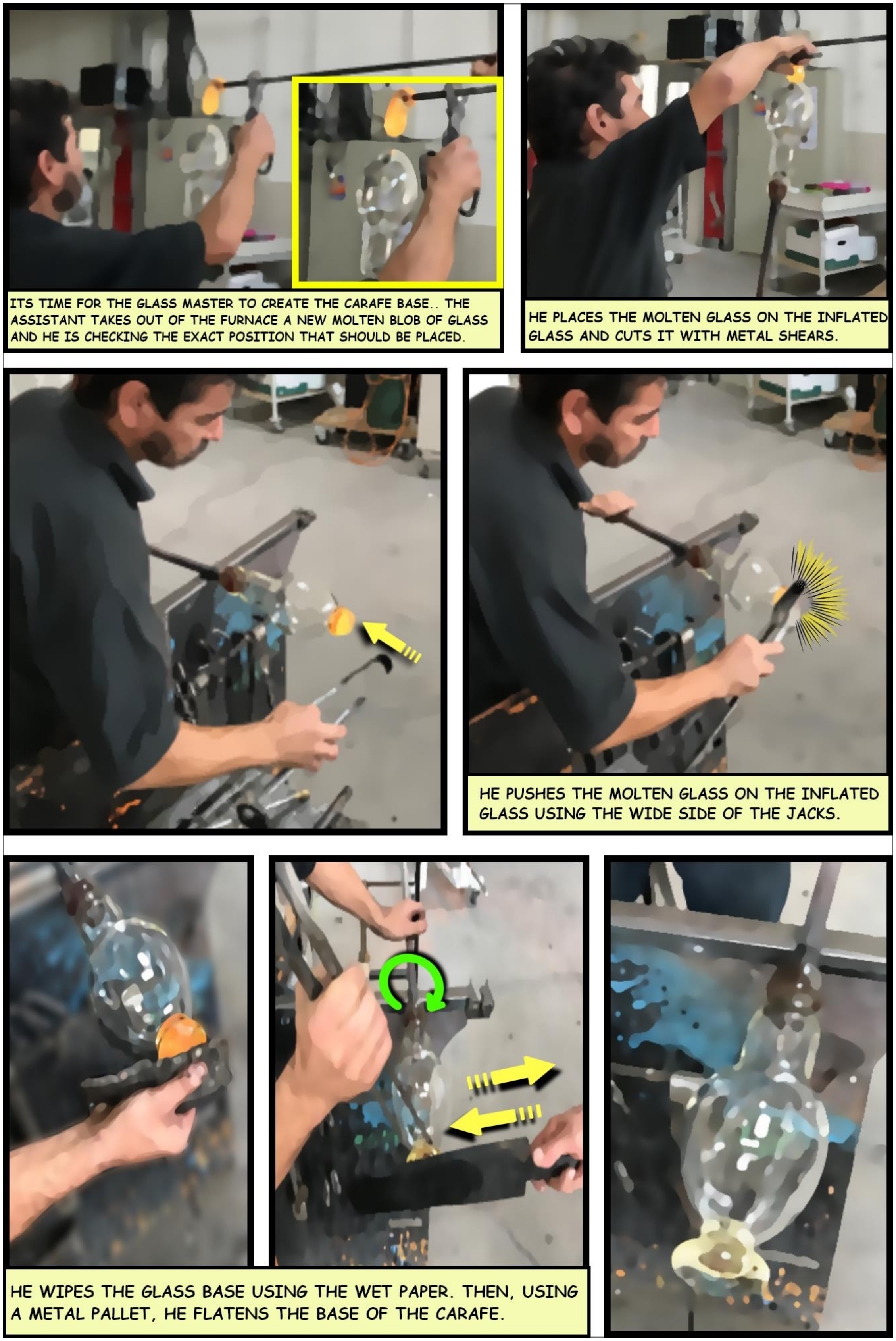


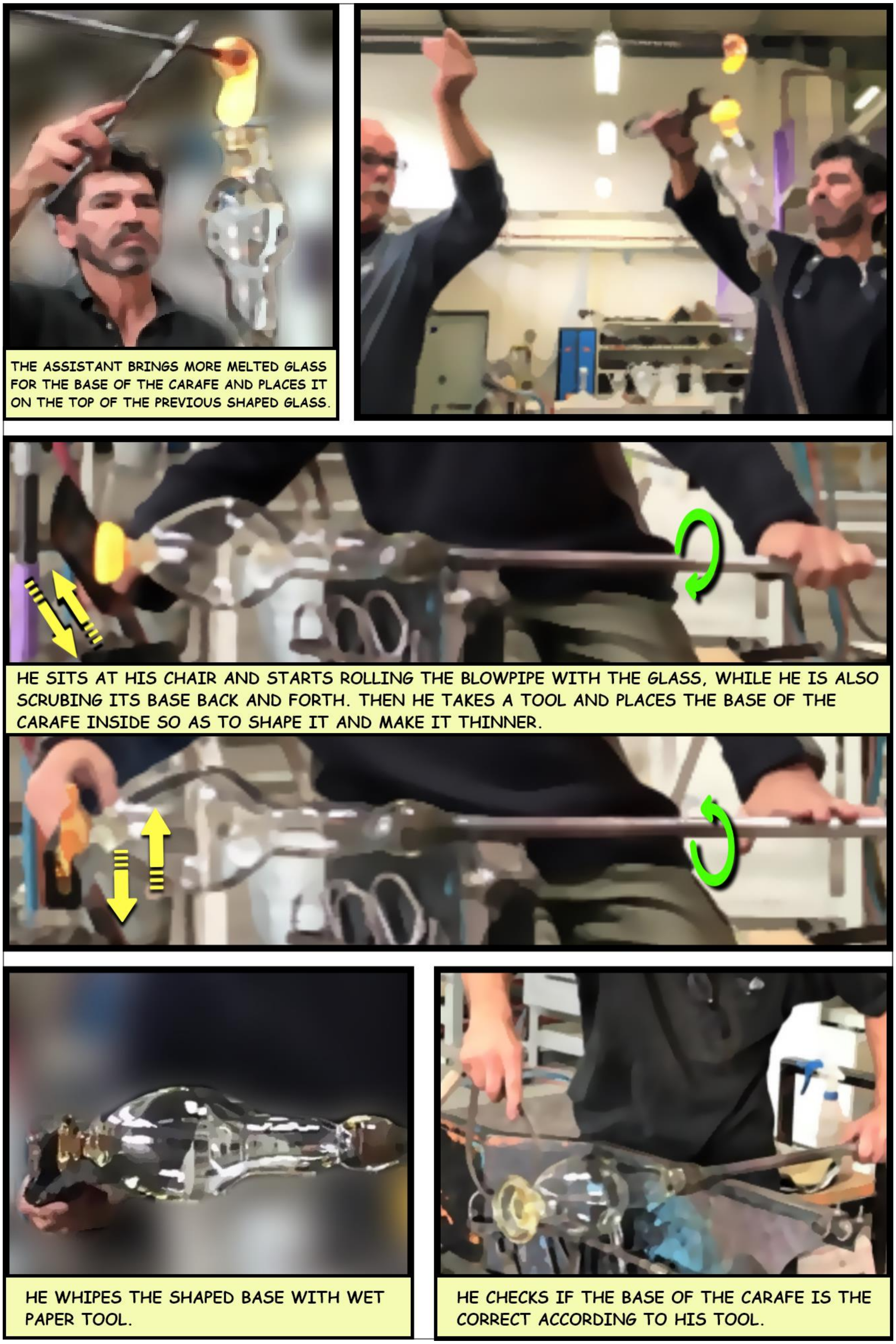
HE USES METAL JACKS TO SHAPE THE GLASS.

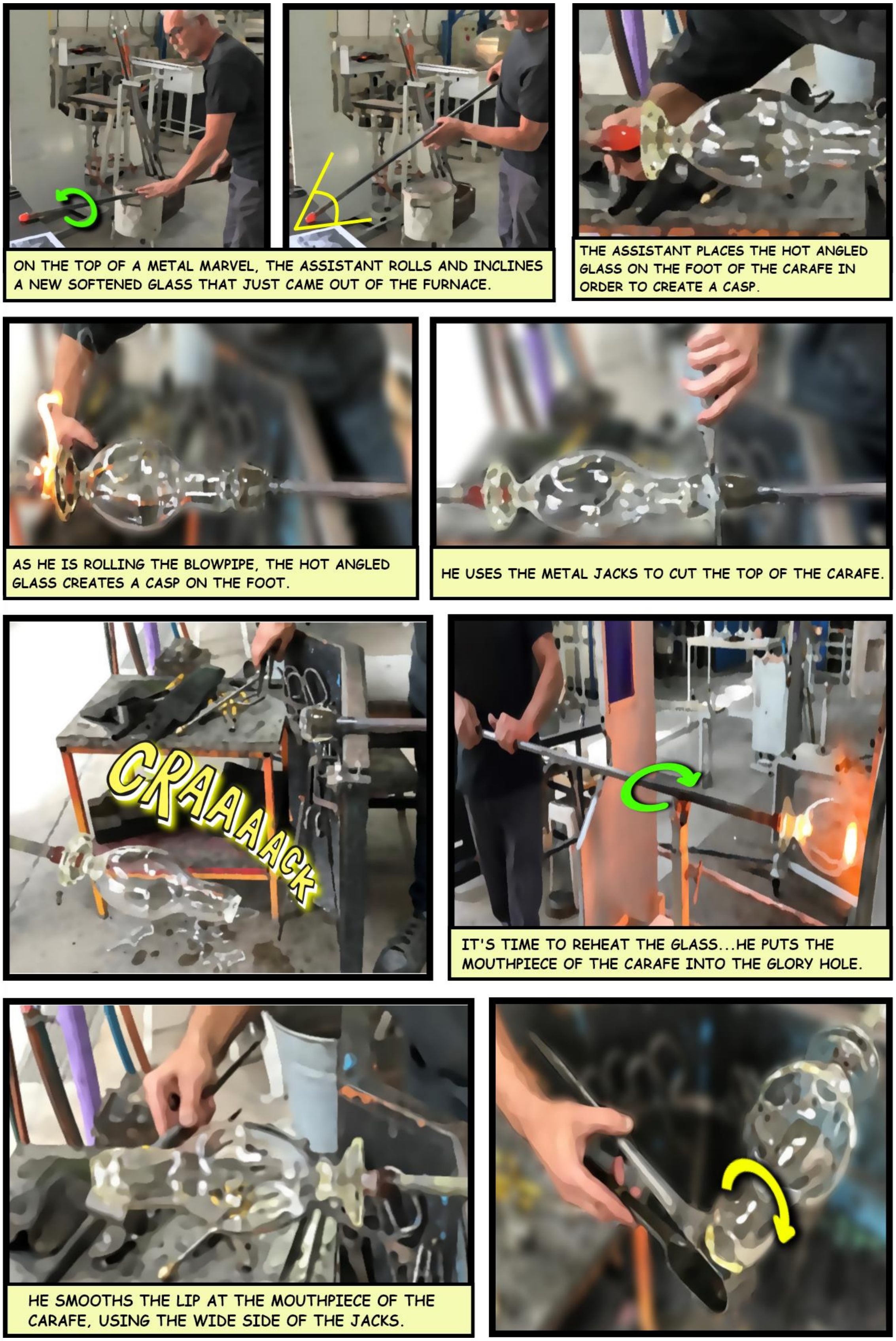
AFTER RE-HEATING THE GLASS, HE BLOWS AGAIN THROUGH THE BLOWPIPE FOR A COUPLE OF TIMES.

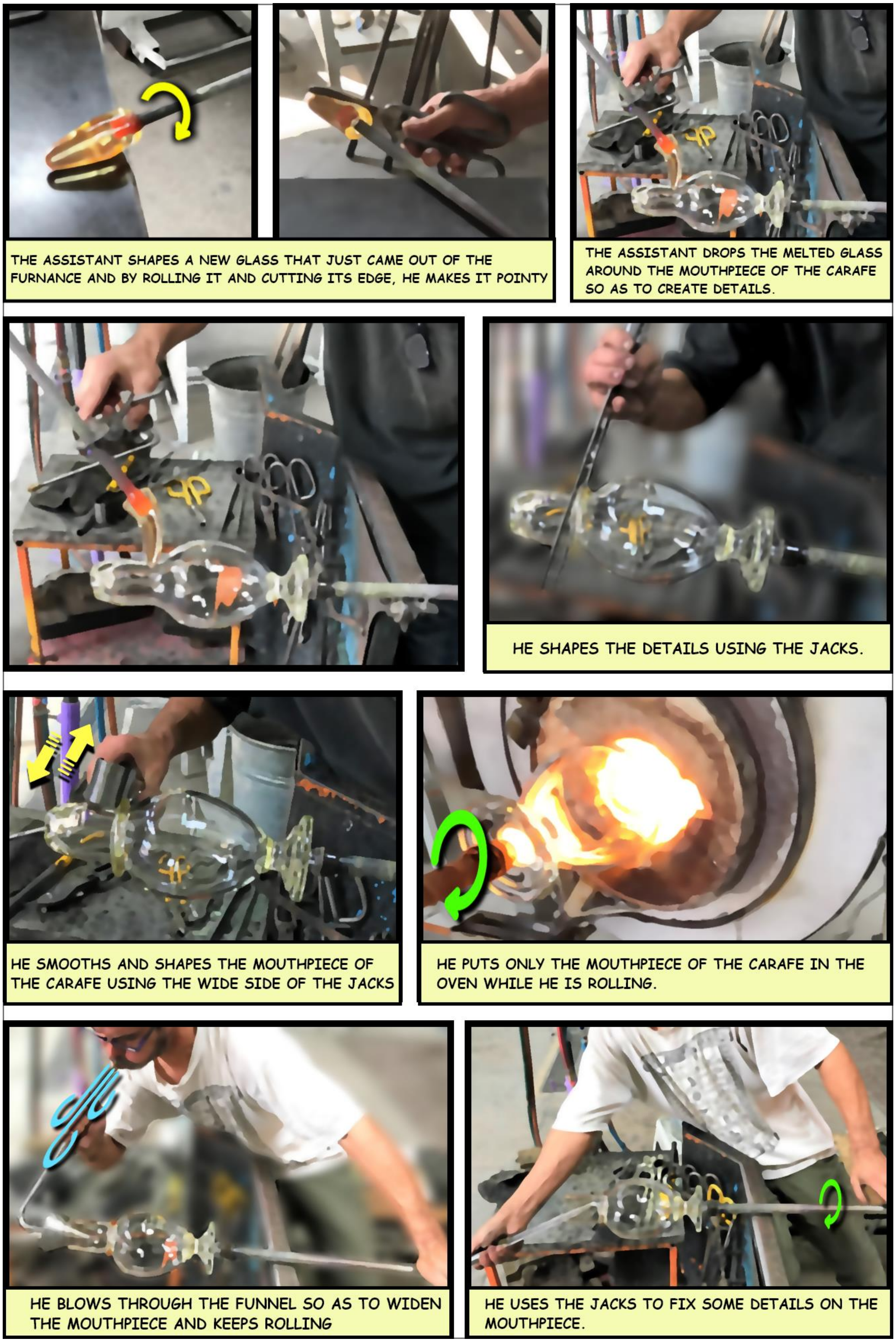
HE CHECKS IF THE GLASS HAS THE APPROPRIATE SIZE ACCORDING TO THE PATTERN SHAPE.

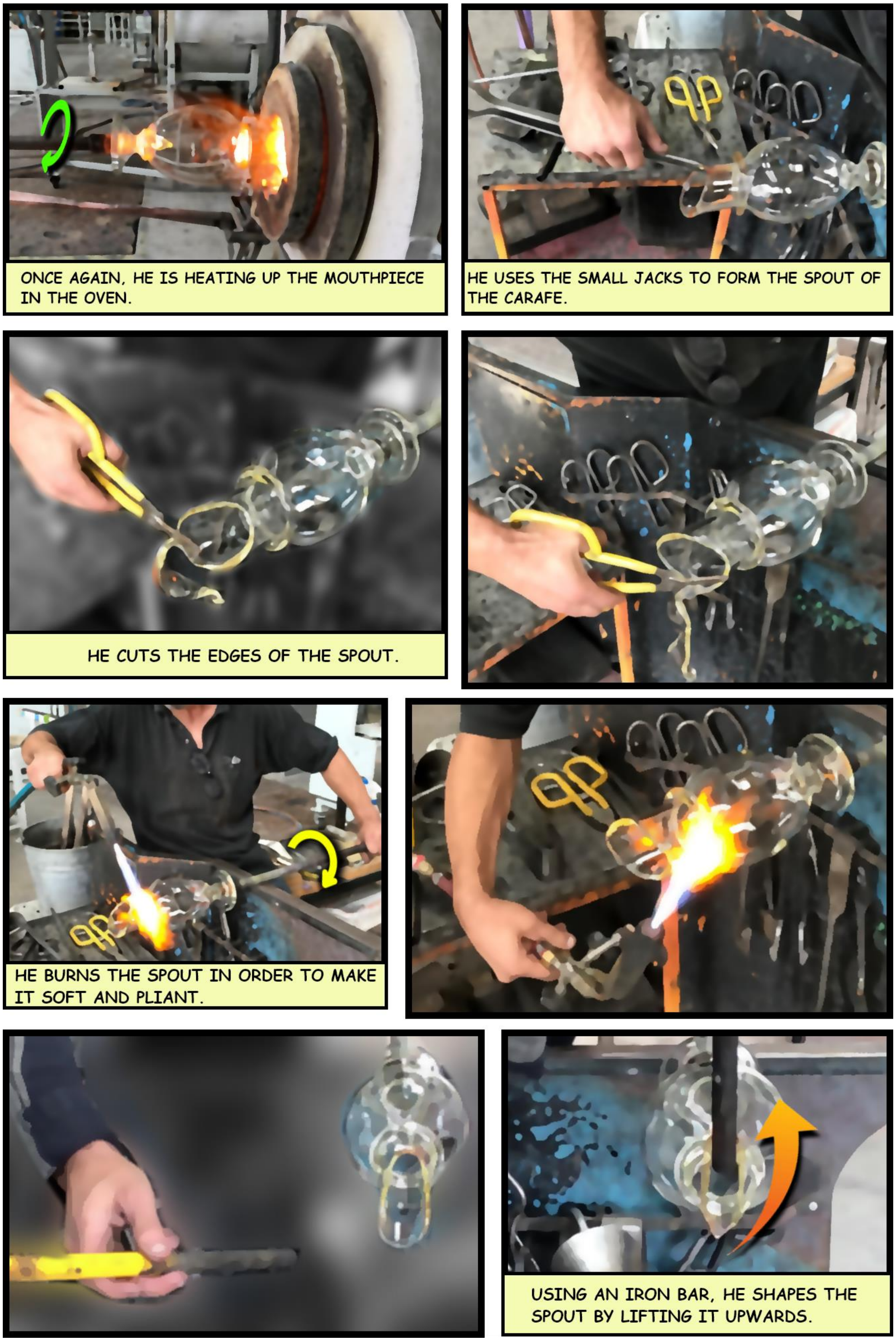
HE BALANCES AND WIPES THE GLASS WITH THE WET PAPER TOOL WHILE HE IS ROLLING THE PIPE.

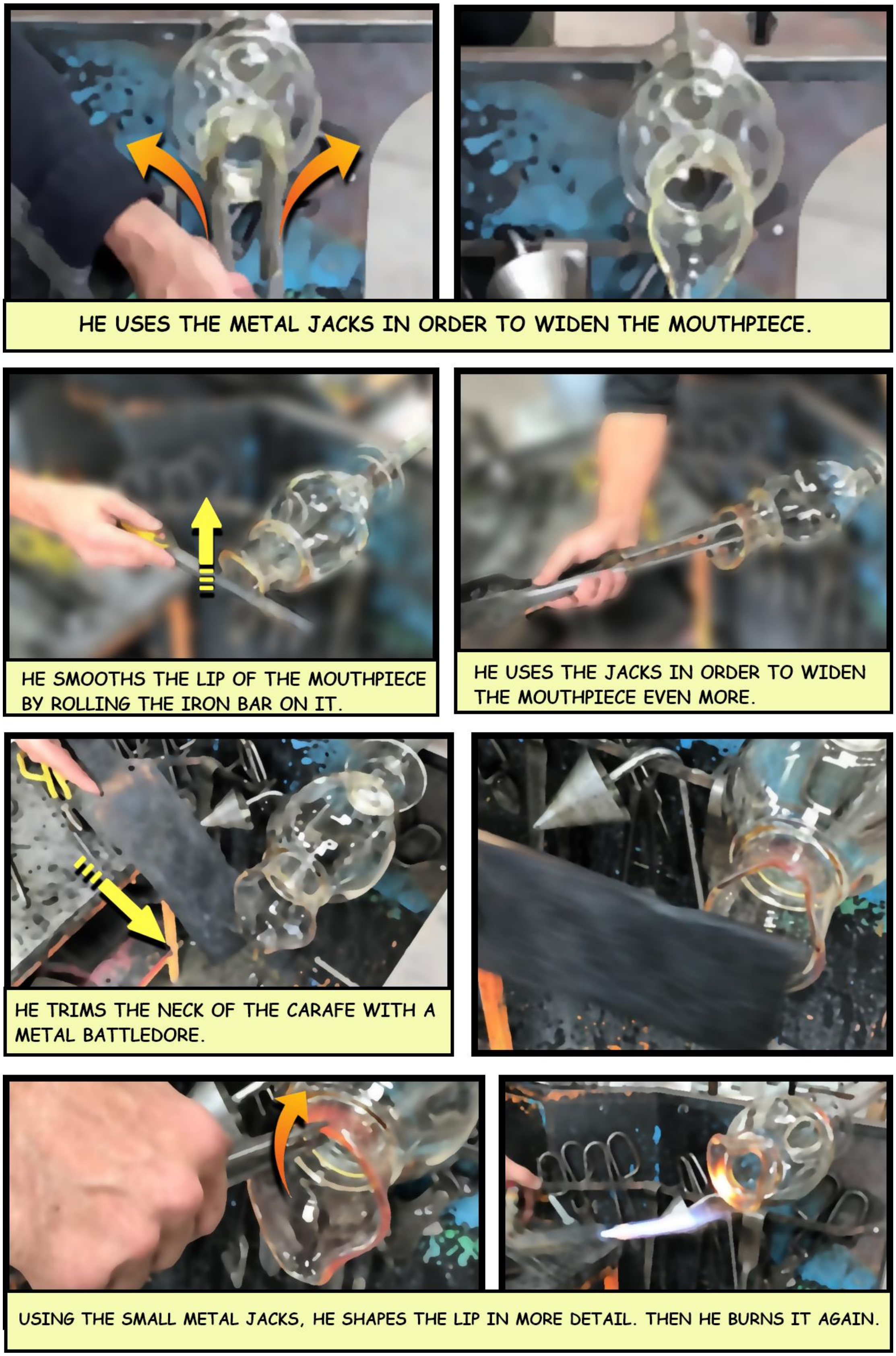

ITS TIME FOR THE GLASS MASTER TO CREATE THE CARAFE BASE.. THE ASSISTANT TAKES OUT OF THE FURNACE A NEW MOLTEN BLOB OF GLASS AND HE IS CHECKING THE EXACT POSITION THAT SHOULD BE PLACED.
HE PLACES THE MOLTEN GLASS ON THE INFLATED GLASS AND CUTS IT WITH METAL SHEARS.
HE PUSHES THE MOLTEN GLASS ON THE INFLATED GLASS USING THE WIDE SIDE OF THE JACKS.
HE WIPES THE GLASS BASE USING THE WET PAPER. THEN, USING A METAL PALLET, HE FLATENS THE BASE OF THE CARAFE.

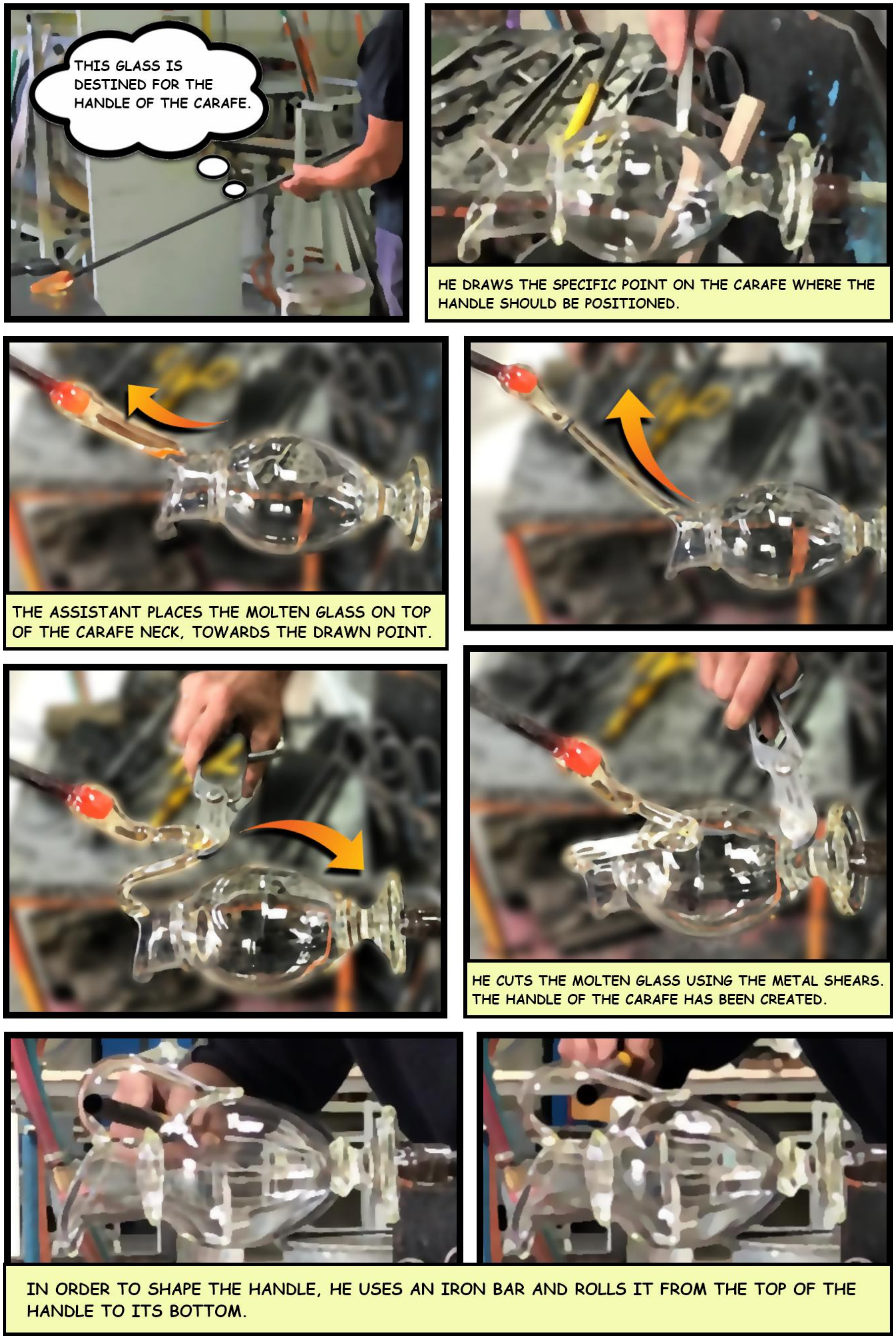

THE ASSISTANT BRINGS MORE MELTED GLASS FOR THE BASE OF THE CARAFE AND PLACES IT ON THE TOP OF THE PREVIOUS SHAPED GLASS.
HE SITS AT HIS CHAIR AND STARTS ROLLING THE BLOWPIPE WITH THE GLASS, WHILE HE IS ALSO SCRUBING ITS BASE BACK AND FORTH. THEN HE TAKES A TOOL AND PLACES THE BASE OF THE CARAFE INSIDE SO AS TO SHAPE IT AND MAKE IT THINNER.
HE WHIPES THE SHAPED BASE WITH WET PAPER TOOL.
HE CHECKS IF THE BASE OF THE CARAFE IS THE CORRECT ACCORDING TO HIS TOOL.

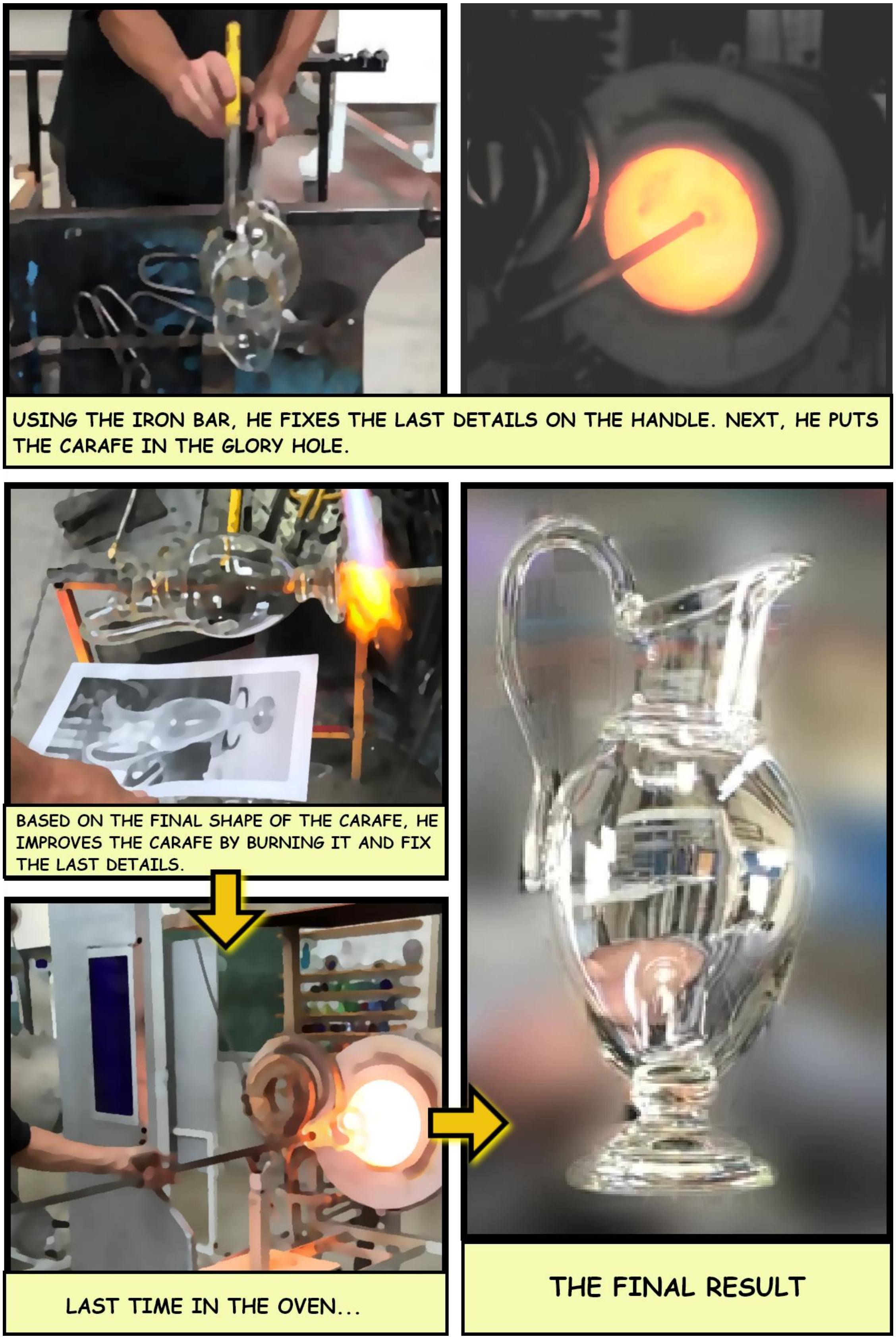

ON THE TOP OF A METAL MARVEL, THE ASSISTANT ROLLS AND INCLINES A NEW SOFTENED GLASS THAT JUST CAME OUT OF THE FURNACE.
THE ASSISTANT PLACES THE HOT ANGLED GLASS ON THE FOOT OF THE CARAFE IN ORDER TO CREATE A CASP.
AS HE IS ROLLING THE BLOWPIPE, THE HOT ANGLED GLASS CREATES A CASP ON THE FOOT.
HE USES THE METAL JACKS TO CUT THE TOP OF THE CARAFE.
CRAAAAACK
IT'S TIME TO REHEAT THE GLASS...HE PUTS THE MOUTHPIECE OF THE CARAFE INTO THE GLORY HOLE.
HE SMOOTHS THE LIP AT THE MOUTHPIECE OF THE CARAFE, USING THE WIDE SIDE OF THE JACKS.


THE ASSISTANT SHAPES A NEW GLASS THAT JUST CAME OUT OF THE FURNANCE AND BY ROLLING IT AND CUTTING ITS EDGE, HE MAKES IT POINTY
THE ASSISTANT DROPS THE MELTED GLASS AROUND THE MOUTHPIECE OF THE CARAFE SO AS TO CREATE DETAILS.
HE SHAPES THE DETAILS USING THE JACKS.
HE SMOOTHS AND SHAPES THE MOUTHPIECE OF THE CARAFE USING THE WIDE SIDE OF THE JACKS
HE PUTS ONLY THE MOUTHPIECE OF THE CARAFE IN THE OVEN WHILE HE IS ROLLING.
HE BLOWS THROUGH THE FUNNEL SO AS TO WIDEN THE MOUTHPIECE AND KEEPS ROLLING
HE USES THE JACKS TO FIX SOME DETAILS ON THE MOUTHPIECE.


ONCE AGAIN, HE IS HEATING UP THE MOUTHPIECE IN THE OVEN.
HE USES THE SMALL JACKS TO FORM THE SPOUT OF THE CARAFE.
HE CUTS THE EDGES OF THE SPOUT.
HE BURNS THE SPOUT IN ORDER TO MAKE IT SOFT AND PLIANT.
USING AN IRON BAR, HE SHAPES THE SPOUT BY LIFTING IT UPWARDS.


HE USES THE METAL JACKS IN ORDER TO WIDEN THE MOUTHPIECE.
HE SMOOTHS THE LIP OF THE MOUTHPIECE BY ROLLING THE IRON BAR ON IT.
HE USES THE JACKS IN ORDER TO WIDEN THE MOUTHPIECE EVEN MORE.
HE TRIMS THE NECK OF THE CARAFE WITH A METAL BATTLEDORE.
USING THE SMALL METAL JACKS, HE SHAPES THE LIP IN MORE DETAIL. THEN HE BURNS IT AGAIN.


THIS GLASS IS DESTINED FOR THE HANDLE OF THE CARAFE.
HE DRAWS THE SPECIFIC POINT ON THE CARAFE WHERE THE HANDLE SHOULD BE POSITIONED.
THE ASSISTANT PLACES THE MOLTEN GLASS ON TOP OF THE CARAFE NECK, TOWARDS THE DRAWN POINT.
HE CUTS THE MOLTEN GLASS USING THE METAL SHEARS. THE HANDLE OF THE CARAFE HAS BEEN CREATED.
IN ORDER TO SHAPE THE HANDLE, HE USES AN IRON BAR AND ROLLS IT FROM THE TOP OF THE HANDLE TO ITS BOTTOM.


USING THE IRON BAR, HE FIXES THE LAST DETAILS ON THE HANDLE. NEXT, HE PUTS THE CARAFE IN THE GLORY HOLE.
BASED ON THE FINAL SHAPE OF THE CARAFE, HE IMPROVES THE CARAFE BY BURNING IT AND FIX THE LAST DETAILS.
LAST TIME IN THE OVEN...
THE FINAL RESULT

The interactive comic book: Based on existing technology by FORTH for interaction with printed matter a demonstration application was implemented that allows users to access more information while interacting with the physical comic book as presented in the following video:

https://youtu.be/UeX3wwQW7WE