Multi-Beam Processing with Individually Addressable Beamlets: Calibration & Data Processing

Alexander Meyer* and Milena Zuric

Fraunhofer Institute for Laser Technology (ILT)
**Corresponding author's e-mail: alexander.meyer@ilt.fraunhofer.de*

Multibeam processing with ultra-short pulsed lasers attracts increasing attention as upscaling technology for laser material processing. While static multibeam approaches for periodic patterns are already entering industry applications, multibeam solutions for the processing of arbitrary geometries are still missing. The demand for high speed, high volume structuring of large surfaces exists, e.g. for molding applications, electronics and photovoltaic, and pushes the development of high-power ultrashort pulsed lasers and the corresponding beam handling systems. To gain the flexibility to process arbitrary structures with a multibeam setup, a flexible, individual control of each beam in the multibeam bundle needs to be realized. This requires significant control of the different modulators typically combined with a galvanometer-scanner based processing setup. A concept for such an adapted processing system is discussed, incorporating acusto-optic modulators for flexible beam switching and a Field Programmable Gate Array for real-time control. A concept to compensate optical aberrations is introduced, and the required algorithms to generate the control commands for the galvanometer scanner and acusto-optic modulators from the provided structure is demonstrated. Finally, the implementation of the real-time scanner-position based switching of the acusto-optic modulators is presented.

DOI: 10.2961/jlmn.2021.02.2005
Keywords: multibeam, ultra-short pulsed laser, kilo-watt laser, micro structuring, control system, galvanometer scanner, scanfield calibration, FPGA, system technology

1. Introduction

1.1. Background

Laser material processing is used as a machining technology for material ablation in various fields of applications such as electronics, photovoltaics, printing and mold fabrication [1–5]. The main benefits offered by laser processing compared to conventional processing are contactless processing without mechanical stress on the workpiece and the wear-free tool. Using ultrashort pulsed lasers with pulse durations below 10 ps also offers very high precision and a negligible heat affected zone in the workpiece due to minimal heat diffusion resulting from the extremely short light-matter interaction times [6,7].

The downside of materials processing with ultrashort lasers is a low material ablation rate, which makes it unsuitable for many industry applications today. Thus, increasing productivity and throughput has been a growing focus in the field of materials processing with ultrashort pulsed lasers. Different approaches to increase productivity will be discussed in the following section. The "MultiFlex" project funded under the H2020 grant by the European Union aims for achieving a substantial increase in productivity by multiple technical improvements: A 1 kW average output power ultrashort laser is developed and combined with a novel multibeam processing optics to distribute the pulse energy across the workpiece for efficient material processing.

This article provides an overview of the new concept of the "MultiFlex" approach and focuses on the principles of the system technology, the control software, the calibration and processing.

1.2. Approaches for increasing productivity

The development of ultrashort pulsed lasers continuously increased the average output power of those lasers up to the kilowatt class in recent years [8,9]. However, the high output power cannot directly be used to increase the productivity of laser materials processing. To achieve the high quality and negligible heat affected zone discussed above, the applied laser fluence needs to be adjusted to a certain fluence value for efficient processing. This optimal fluence is dependent on the processed material and is typically a factor of e^2 or approx. seven times the ablation threshold fluence of the material under investigation [10]. In combination with a small laser focus diameter of 20 µm to generate small feature sizes, typically 10 W of average output power can be applied for metal processing with a single laser spot using a conventional galvanometer scanner technology.

Thus, to efficiently convert the high average output power of multiple hundred watt of modern ultrashort pulsed lasers into ablation products and a high removal rate, the output power needs to be distributed homogeneously across the workpiece. This can generally be achieved in two ways: Firstly, by increasing the repetition rate of the laser pulses and using high-speed beam deflection systems to realize a moderate pulse overlap and separate successive pulses, e.g. with a polygon scanner [11]. Secondly, by splitting the high energy pulses into multiple beamlets to process multiple

workpieces or multiple positions on the same workpiece in parallel [12].

1.3. Multibeam processing of a single workpiece

Processing a single workpiece with multiple laser beamlets corresponds to the machining of one workpiece with multiple tools at the same time. In this way, each beamlet can operate at the optimized fluence while the total laser output power applied for processing can be scaled by the number of applied beamlets.

For splitting of the laser beam into multiple beamlets, phase masks such as a diffractive optical element (DOE) are commonly used as nearly arbitrary beamlet patterns with almost any number of partial beams can be created. A typical setup for the optical system of a static multibeam setup utilizing a DOE for beam splitting is shown in Fig. 1. The bundle of beamlets is parallelized by a relay optic and a mask is used to filter scattered radiation from higher diffraction orders, which does not contribute to the ablation process. A second relay optics focuses the beamlets and directs the beamlet bundle into a standard galvanometer scanner.

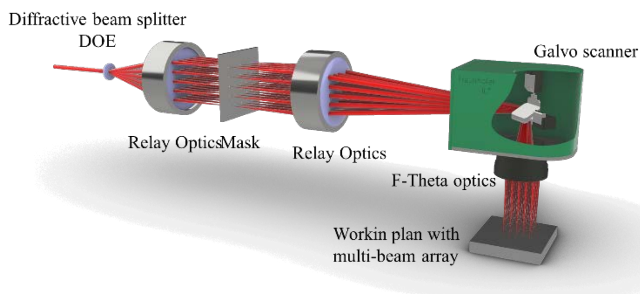

Fig. 1 Schematic optical setup for a static multibeam pattern with a galvanometer scanner.

A static beam pattern like this can already provide a significant increase of productivity for applications requiring strictly periodic structures with a periodicity equal to or smaller than the pitch of the beamlets[12]. However, the generation of arbitrary structures is not possible. Due to distortions induced by the optical system, a position error occurs with increasing distance from center of the scan field. To maintain the required machining precision, the size of the beam pattern as well as the scan filed can be confined [13]. Confining the beam pattern and maintaining the overall number of beamlets results in an increased power density, which might cause the aforementioned heat related issues such as a decrease in surface quality [14].

The MultiFlex project addresses these drawbacks by introducing an array of acusto-optic modulators (AOMs) in the beampath of the optical setup in front of the mask to enable individual switching of every single beamlet [15,16]. This allows for the generation of arbitrary beam pattern and surface structures. Additionally, the adjustment of the switching pattern of individual beamlets allows for compensation of optical distortions. Hence, the restrictions to the size of beam pattern and scan field can be removed. The machine will include a 1 kW average output power laser with a pulse duration of less than 1 ps.

The optical concept for the MultiFlex machine is shown in Fig. 2. 64 beamlets will be arranged in an 8x8 beam pattern with a nominal pitch of 4 mm between neighboring beamlets. For beam deflection a standard galvanometer scanner is used. A detailed discussion on the development of the optical system has been published in [17].

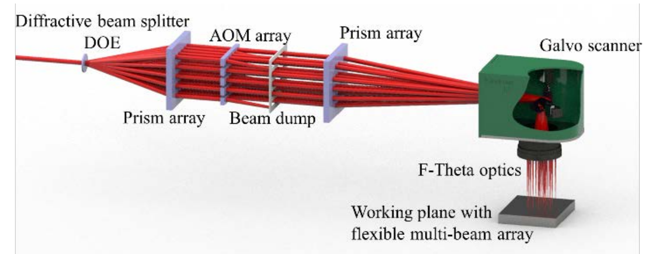

Fig. 2 Principle of the optical concept in the MultiFlex project.

This article will focus on the individual beamlet calibration, data processing and the data flow in the machine control system.

1.4. Control system overview

A schematic of the control system is shown in Fig. 3. The data flow in the various stages is as follows:

(1) A bitmap image describing the structure to be created is inserted into the “JobCreator” software. The grayscale values indicate the number of layers to process at any given pixel. Further information of such as resolution (mm / px) and process strategies are provided by the user.
(2) The “JobCreator” software splits the provided image into smaller patches for each beamlet. The software creates a job file using the OpenVectorFormat [18] containing
    a. Vector data for the scanner
    b. Switch pattern for each AOM channel
    c. Axis positions for focus adjustments & movements of the xy-stage for large structures.
(3) The control software on the machine loads the job data and sends scanner vector pattern and AOM switching pattern for the first patch to the scanner and Field Programmable Gate Array (FPGA), respectively.
(4) The control software moves the axis to the appropriate position.
(5) Processing of the patch is started.
(6) The scanner and FPGA perform a handshake for synchronization.
(7) The scanner processes the provided vector pattern. In real time, the FPGA observes the actual position of the scanner through a sniffing extension of the scanner controller and switches each AOM channel on and off at the appropriate scanner positions.
(8) After processing of the patch is finished, the process restarts at (3) for the next patch, until the complete job is processed.

As the whole beam pattern is moved by the galvanometer scanner, the single beamlets cannot be moved independently from each other. The flexibility required for arbitrary processing is achieved through the independent switching of the AOMs. This implies that with this approach, any processing is pixel-based, and any input structure to be processed needs

to be translated to a digital, pixel-based on / off switching pattern for the AOMs.

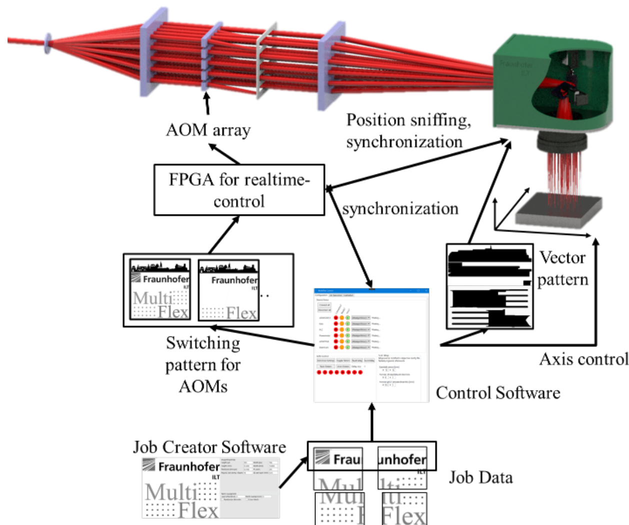

Fig. 3 Schematics of the MultiFlex control system.

Technical specifications of the components used are:

- Newson 3G scanner and a CUA32 controller with sniffing extension
- Avnet UltraZed-EG SOM with Xilinx MPSoC
- 8x custom 8-channel AOMs by AA Optoelectronics

2. Calibration

Fig. 4 shows the characteristic field distortion of a galvanometer scanner in combination with an F-Theta lens. Usually, those distortions are corrected by creating a correction table which is uploaded into the scanner controller [19]. The scanner controller thus automatically adapts the scanner movements and compensates the distortions.

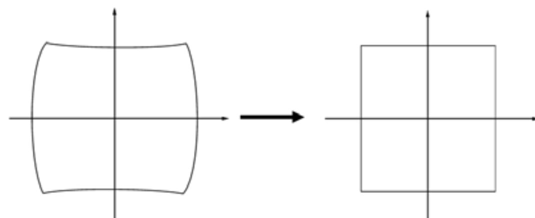

Fig. 4 Left: Characteristic distortion pattern caused by a galvanometer scanner in combination with an F-Theta lens when marking a square.

This straightforward process is not applicable for multibeam systems that guide multiple beamlets through a single scanner. In this case, the beamlets are located at different positions in the scanfield, resulting in a different distortion and displacement for every single beamlet. Furthermore, there is also a rotational distortion that is usually not relevant for single beam processing with circular beams. Here, rotational distortion introduces skew in the beam pattern during multibeam processing, as visible for the deflected patterns in the corner of the scan field in Fig. 5.

Static multibeam systems usually avoid or reduce distortions of the beam pattern by correcting the center of mass of the beam pattern with a scanner correction table and only use small deflection angles or reduced scan field sizes, as the distortions are negligible near the scan-field center, as can be seen in Fig. 5. Furthermore, static multibeam patterns are usually very compact with a typical pitch of less than 1 mm between neighboring beamlets to avoid positioning errors exceeding the precision requirements for the application (see Fig. 6) [13,20].

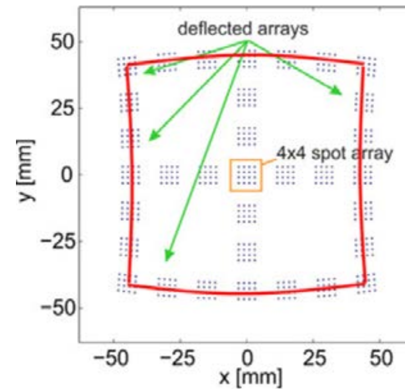

Fig. 5 Distortion effects simulated for a 4x4 multibeam pattern. Taken from [20].

In contrast, the MultiFlex project targets an 8x8 beamlet pattern, with a large beamlet pitch of 4 mm. As shown in Fig. 6, the spot position error for such a configuration would be well above a typically acceptable precision tolerance of 10 µm for a setup with a 100 mm focal length of the F-Theta lens.

Since the largest distance between two beamlets in this pattern is around 40 mm, a calibration only for the pattern as a whole is not suitable. Thus, a method for individual calibration of each beamlet is required. This in turn eliminates the option to perform the correction by adapting the scanner movements, since the scanner always moves the entire pattern. Instead, calibration is done by adapting the switching pattern for the AOMs controlling the individual beamlets.

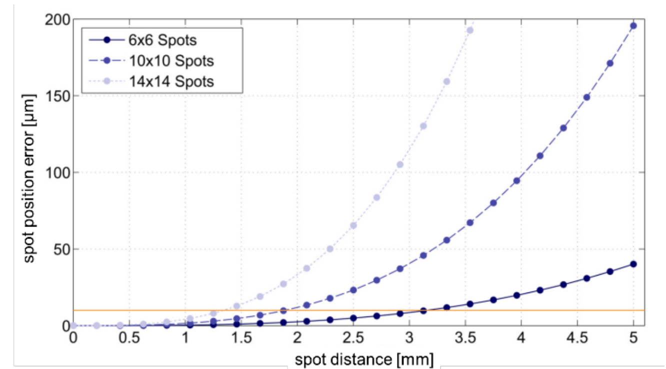

Fig. 6 Spot position error for three different pattern configurations ranging from 6x6 spots to 14x14 spots for a f-theta lens with 100 mm focal length. The orange line indicates a typical precision requirement of 10 µm. Adapted from [13].

2.1. Automated measurement setup

Typically, calibrating a single beam setup to create a correction table requires measuring of at least 9 positions in the scan field and recording the deviations of intended and actual position. This can easily be done by marking e.g. a crosshair at each of the 9 positions on a workpiece and measuring the deviation with a microscope by hand.

However, calibrating 64 beamlets in this fashion would require at least 576 measurements and thus a manual workflow is not suitable. Instead, an automated calibration was developed. For the measurement of the position, a Basler acA3800-10gm GigE Vision camera is mounted in position of the workpiece and moved by the axis system. To reduce the laser power on the camera sensor, the laser beam is reflected of the anti-reflectively coated side of two wedge plates in series before entering the camera. Each reflection removes 99.5 % of the laser power, reducing the laser power illuminating the camera's sensor enough to avoid damage of the camera. The non-reflected part of the laser power passes through the wedge plates into a beam dump. The wedge shape also ensures that back reflections from the rear surface of the plates are not guided onto the camera, avoiding ghost images. Furthermore, using reflective optical components, as opposed to transmissive components such as neutral density filters, ensures that there is no beam displacement other than those of the multibeam optics.

To determine the position of the beam on the camera chip, a custom camera software is applied. The beamlet's position in the acquired image is determined by first calculating the approximate position of the spot using the central moments of the image. Afterwards, a gauss fit is performed around this approximate position to get the exact spot beamlet position with respect to the image coordinates. The software uses the GenICam standard to communicate with the camera and is remotely controllable through a gRPC interface.

The flow-control for measuring the position of one beamlet is handled by the machine control software with the following steps: (1) move camera (on xy-stage) and scanner mirrors to target position → (2) enable single beamlet by switching on the laser and corresponding AOM channel → (3) request position measurement through gRPC from camera software → (4) switch laser & AOM channel off. Those four steps are repeated for each of the 576 measurements.

The accuracy of the calibration setup is validated by measuring the resolution of the camera, which is specified as 598.8 px/mm by the manufacturer. The position of a laser spot on the camera was measured and the camera was moved 1 mm with a high precision axis before the position was measured again. This process was repeated 50 times and the measured resolution is (598.72 ± 0.62) px/mm. The measurement uncertainty is right in range with the expectation for the camera's pixel size of 1.67 µm.

Fig. 7 shows the field distortion for a sample measurement, displaying the typical displacement pattern expected form a galvanometer scanner and a f-theta lens.

Even with the fully automated system, the calibration measurements for all 64 beamlets takes around 1 hour.

The calibration is applied to the job data as a post-processor after the generation of the job (see next section). For this, the calculated distortion matrix for a beamlet is applied to the part of the image that will be processed by this beamlet (BeamletTiles, see section 3.2) in a post processing step. By distorting the input data to the machine in this manner, the distortions of the scanfield should be compensated. A final demonstration of this concept is the subject of ongoing work.

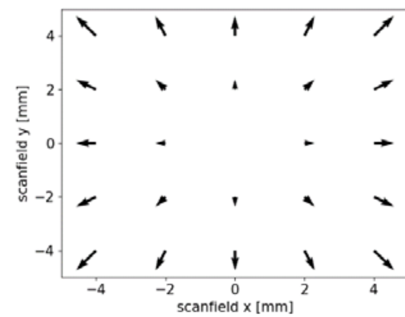

Fig. 7 Measured field distortion for an uncalibrated galvanometer scanner with f-theta lens. Magnitude of the arrows is increased for better visualization of distortions.

3. JobCreator Software

3.1. Architecture

For efficient and flexible generation of a scan job to process a variety of structures with the multibeam pattern, a flexible software approach is needed to enable quick integration of different processing strategies. Job creation and job execution take place in two separate applications. This enables job creation on a different computer than the one in the machine. Two sets of data are exchanged between the two applications:

- The machine software provides a beamlet pattern XML-configuration file to load into the JobCreator software. This dataset contains
  - Number of beamlets
  - Pattern configuration (1x8, 8x8 etc.)
  - Nominal pitch between beamlets
  - Result of the above calibration for each beamlet
  - Scan field size
- The JobCreator software saves an OpenVector-Format file containing
  - Metadata on the job (source data, processing strategy, author etc.)
  - Axis positions to reposition the XY-stage between patches
  - Axis positions to adjust the focus between processing layers
  - Scanner vectors
  - Bitmask for position based AOM switching

The JobCreator software is built in C# with a graphical user interface using the WPF library. At the moment, only a bitmap segmentation algorithm to create a job is implemented (see 3.2). However, the application was designed using a Model-View-ViewModel pattern and is very modular, so different processing strategies can be implemented easily.

3.2. Basic bitmap segmentation algorithm

All structures to be processed in the MultiFlex-project are based on grayscale-bitmap images. Thus, the primary focus of the implementation of job creation algorithms is a robust and efficient algorithm to generate a job from a bitmap image. In the grayscale images, the number of layers to be structured at any given position is encoded through the grayscale value of the corresponding pixels. In between, linear scaling is applied to create e.g. 400 processing layers from the maximum 255 image layers in an 8-bit grayscale image.

Besides the image file, additional input required for job creation from the user is:

- Target resolution (mm/px)
- Number of layers to be ablated (processing layers)
- Number of layers in one “layer package” (the focus position only gets adjusted for each layer package, not for every single layer. This speeds up the process significantly.)

For processing, each process layer of the image needs to be segmented into smaller patches for each beamlets. We choose a multi-level segmentation approach and, for every layer, the processing algorithm sequentially

- splits the image into “AxisTiles” – pieces of the image that can be processed by the scanner alone, without movement of the axis in x or y.
- Each AxisTile is split into “ArrayTiles“ – pieces of the AxisTile that the beamlet pattern (array) can process at once
- The ArrayTile is split into “BeamletTiles” – the piece of the ArrayTile that one individual beamlet will process.

For every ArrayTile, the scanner movement is generated in such a way that each beamlet reaches the starting position of its neighboring beamlet, so that the whole ArrayTile is processed without gaps (Fig. 8). Furthermore, for each BeamletTile in every ArrayTile, a position map is calculated. This position map provides the information when the beamlet corresponding to the BeamletTile needs to be switched on.

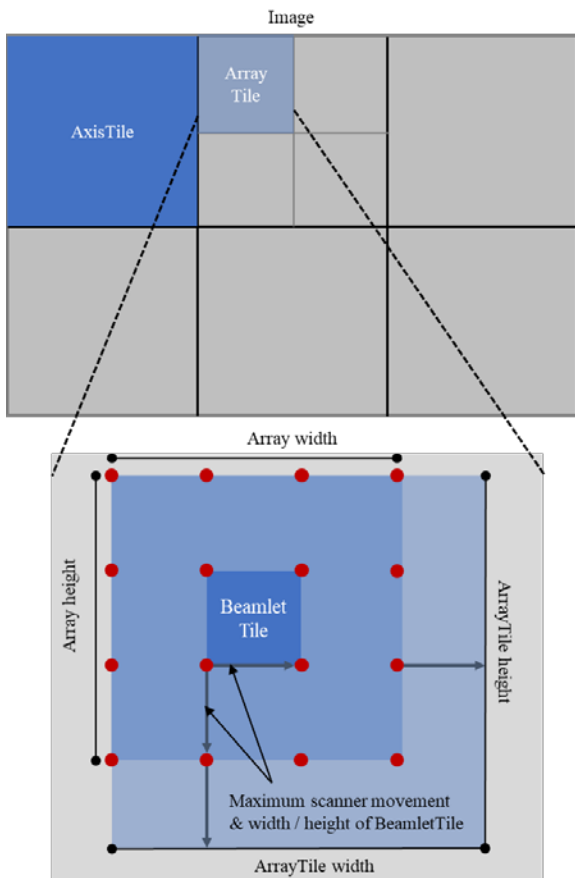

Fig. 8 Concept of the splitting of the image into tiles. The red dots represent individual beamlets. For illustration purposes, only a 4x4 array is shown.

Due to imperfections in the DOE and other optical components, as well as non-preventable tolerances while adjusting the optical module, the placement of the beams in the beamlet array may be slightly irregular. This causes some tiles to slightly overlap with their neighboring tiles since the scanner needs to move the maximum distance between any neighboring tiles in the array to process the complete workpiece, as illustrated in Fig. 9. To avoid parts of the structure to be processed multiple times due to these overlaps, the complete area of a tile (including overlap to neighbors) is blacked out in the source image. Thus, when the part of the image to be processed in the next tile is extracted from the source image, the part already processed by the first tile in the overlapping section does not contain any structure for processing anymore.

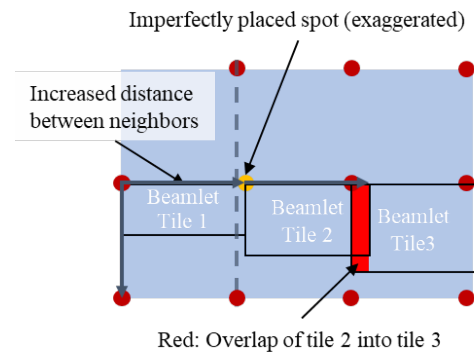

Fig. 9 Illustration of the problem caused by imperfections in the beamlet spacing within an ArrayTile. The height of the tiles is only different for illustration purposes – in the process, they all reach the next row of neighboring beamlets below.

Mapped to the OpenVectorFormat data structure, a ArrayTile corresponds to a VectorBlock – it contains the scanvectors for the size of one BeamletTile, and the switching pattern for all AOMs, so the complete ArrayTile gets processed. An AxisTile corresponds to one WorkPlane in the job data structure, because the axis coordinates differ for each AxisTile.

The processing pipeline, as the whole program, is designed to be highly flexible. By simply adjusting the XML array configuration, jobs for a completely different array configuration can be created. Fig. 10 shows a job created for a 1x8 beamlet array.

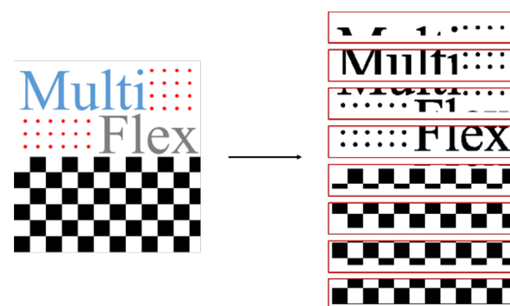

Fig. 10 Bitmap image fitting into a single ArrayTile split into BeamletTiles for processing with an 1x8 beamlet pattern.

A special challenge in the implementation of this approach is the handling of the multitude of different coordinate systems. For one, there is the workpiece coordinate system, but also the coordinate system of each AxisTile and the

positions of the ArrayTiles inside, and the coordinates of each ArrayTile with the BeamletTile coordinates inside. Additionally, image coordinates usually use an inverted y-axis to address the pixel coordinates. Getting the coordinates correct in all structure levels and matching it with the corresponding parts of the image is one of the main challenges.

3.3. Reducing the effect of patch boarders

The following sections will cover enhancements to the previously introduced algorithm to address the risk of quality degradation of the finished workpiece due to visible patch boarders.

3.3.1. Visible transitions at patch boarders

Visible transitions on workpieces processed by scanning laser system can occur whenever the processing of a connected area cannot be executed in one continuous scanning motion. The most common scenario for this is stitching or transition between neighboring scan fields during processing of large areas. Due to a positioning error of the axes and / or imperfect calibration of the scanner, a so-called stitching-error can occur, as shown in Fig. 11.

Due to the multi-level tiling approach described in 3.2, a multitude of patch boarders are present in any structure processed by the multibeam-system. We now present two approaches to reduce the impact of patch boarders, applicable for different types of structures.

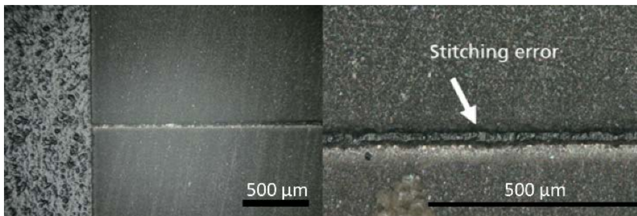

Fig. 11 Example of a stitching error. The two areas should join seamlessly.

3.3.2. Shifting patch boarders for larger, continuous structures

The first approach shifts the position of the tile borders between the layers of the process. This prevents stitching errors to “stack up” over multiple layers and become more apparent. To preserve the modularity of the image processing pipeline, this addition is implemented as a pre-processor. Before the image data for a layer gets segmented into tiles, some padding is attached to the image, as illustrated in Fig. 12. The image with the padding attached is then processed by the segmentation algorithm and divided into tiles. Since the size of the image + padding is the same for every layer, number and size of the tiles remain unchanged, but the positions of the tile borders vary.

To keep the different layers aligned, the link between the (0,0) pixel of the whole image and the workpiece coordinates needs to be modified. E.g., for the Layer 0 in Fig. 12, the (0,0) workpiece position is matched to the [0,0] pixel position of the image layer. For Layers 1, 5 and 10, the (0,0) workpiece position is at different pixel-positions [a,b] with a,b > 0. An example for the output of this algorithm is shown in Fig. 13.

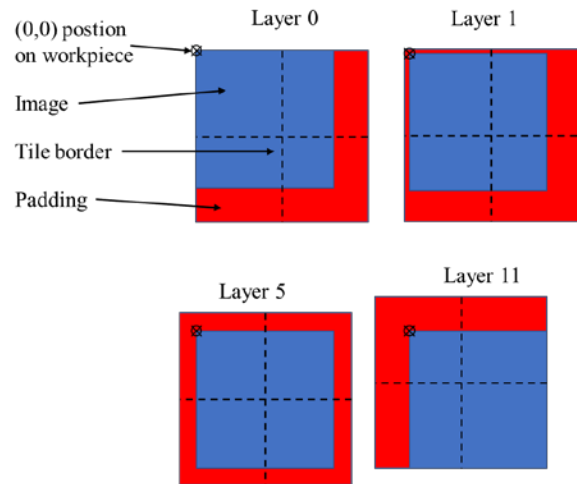

Fig. 12 Illustration of shifting tile borders to different positions by using padding around the image. Tile borders are shown as dashed lines.

A significant advantage of this approach for mitigating visible transitions is that it works on all types of structures, since it does not rely on gaps in the structure, as does the “smart segmentation” approach that follows. This is frequently used in laser processing. A downside of this approach is that there are still transitions between the patches on a layer basis Therefore, transitions form the last layer processed will still be visible. Moreover, it is not applicable to single layer processes, e.g. in thin film processing.

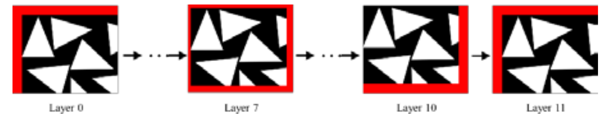

Fig. 13 Shift of the image layer generated on a sample image by the boarder shifting addition.

3.3.3. Smart segmentation – avoiding transitions for discrete structures

The above strategy is agnostic of the structure being processed. This has the advantage that it is applicable to any structure. The disadvantage is the occurrence of visible transitions on the top layer.

Fig. 14 Processing small parts of a structure element reaching into a neighboring tile by allowing an overlap between the processing regions of the tiles. Left: Without overlap. Hard cut-off of the processing at tile borders. Center: Mask of the processing area for each BeamletTile. Right: Adapted processing area mask to avoid cut off of the triangle structure.

As an alternative approach, we introduce an overlap of the processing areas of neighboring BeamletTiles to allow a structure which was previously cut off at the border of the BeamletTile to be processed completely within one of the

tiles, as illustrated in Fig. 14. This smart segmentation approach is applicable specifically for structures containing small, discrete structural elements. The size of these elements should not exceed the size of a BeamletTile.

Since it is necessary to modify the BeamletTiles when they are generated, it is not possible to integrate this feature as a pre- or post-processor to the segmentation process. Instead, it is directly implemented into the segmentation algorithm described in 3.2. For the following illustrations of the algorithm, the sample image with white triangles on a black background will be used to illustrate the process. The white triangles are the structure that shall be processed.

The segmentation of the complete image into Axis- & ArrayTiles is not modified. Starting off with an ArrayTile from the second step in the segmentation process, Fig. 15 (bottom) shows the state of the ArrayTile after some BeamletTiles are generated.

Now, the generation of the next BeamletTile with the indices (n,m) will be shown in detail. The tile and its neighbors are shown in Fig. 16. As mentioned above, the basic idea of this approach is to allow for a (user definable) amount of overlap of all BeamletTiles with their direct neighbors. Thus, the tile area with overlap for the (n,m)-tile is copied to a new image as shown in Fig. 15 (right).

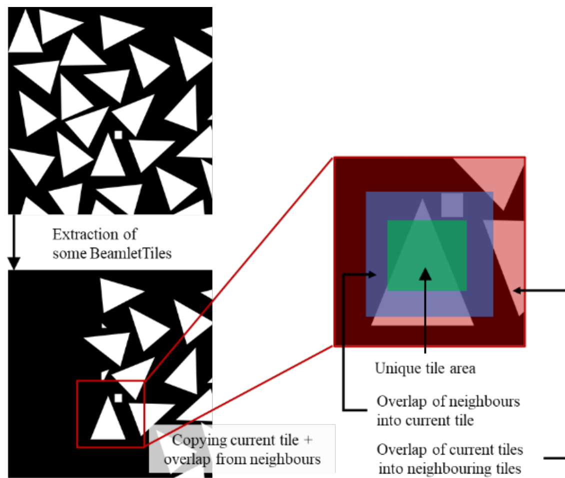

Fig. 15 Top: Complete ArrayTile before segmenting into BeamletTiles.
Bottom: ArrayTile after some BeamletTiles have been generated - everything that is already included in a BeamletTile is blacked out in the ArrayTile.
Right: Copied segment of the remaining ArrayTile for generation of the (n,m) BeamletTile

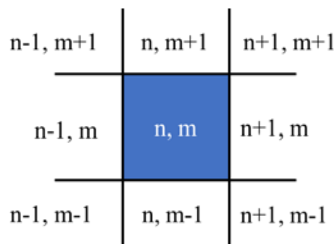

Fig. 16 Current tile (n,m) with its direct neighbors.

As it will be shown later, the algorithm works in such a way on each tile that it leaves only what needs to be processed by a following tile (explicitly the top and right side neighbors: (n-1, m+1), (n, m+1), (n+1, m+1), (n+1, m), (n+1, m-1)) in its overlap area. This means for the current tile (n,m) that everything in the "unique tile area" as well as any structure still present in the overlap area of the preceding tiles ((n, m-1), (n-1, m-1), (n-1, m)) needs to be processed in any case. Thus, the image is divided into the two sections shown in Fig. 17.

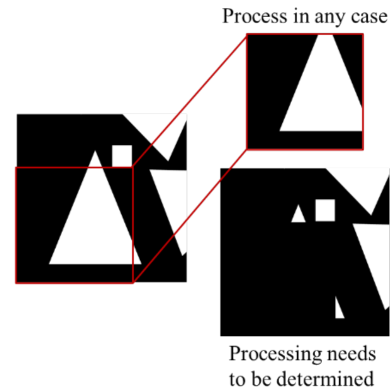

Fig. 17 Division of the (n,m)-tile into the part that needs to be processed in any case (left over for processing by preceding tiles) and the area of overlap with neighboring tiles that still needs to be processed.

Afterwards, the process is continued with the lower image of Fig. 17. It is now evaluated one element at a time if this element should be processed with the (n,m) tile or be deferred for processing with a following tile. Those deferred elements would then appear in a lower left section of one of the following tiles and thus fall into the "process in any case" area of this following tile. The process of individually inspecting the different structure elements is shown in Fig. 18. Starting from the "processing needs to be determined" area shown at the top of Fig. 18, the separated structure elements are identified by a contour detection algorithm provided by the OpenCV library. The decisions in which tile the individual elements should be processed are as follow:

Elements (1) + (5): Element is in contact with the area processed in any case, likely connected to a structure inside this area → process in this tile to avoid transition.

Element (2): Element is not connected to area processed in any case or the outside borders → stand-alone structure element completely contained inside the overlapping area → center of mass is closer to the center of the (n,m) tile than any other tile → process with this tile.

Elements (3) + (4): Element is connected to an outside border, likely connected to a structure in a neighboring tile → do not process in this tile, defer to neighboring tile.

Now, the remaining structure elements in the overlapping area are merged back together with the "process in any case" part of the tile, resulting in the final (n,m) BeamletTile that will be processed by the (n,m) beamlet in the beamlet pattern, as illustrated in Fig. 19.

The last step is to remove the sturctures that are processed within the (n,m) tile from the ArrayTile, since the generation of the following BeamletTiles will use the ArrayTile as an input again.

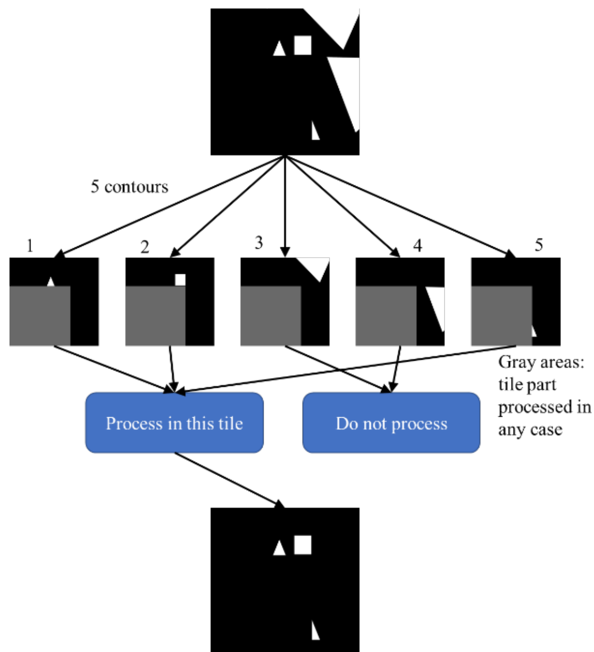

Fig. 18 Evaluating which structure element should be processed in this BeamletTile and which is deferred to a following BeamletTile.

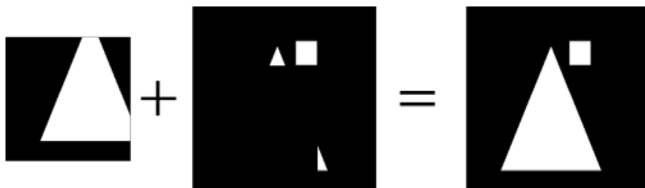

Fig. 19 Merging the "process in any case" part of the tile with the structures from the overlapping area.

As can already be seen in Fig. 19, the resulting BeamletTile contains only complete, non-interrupted structures. With a sufficiently large overlap, any structure can be processed without transitions (if it fits in the scan field). However, taken to the extreme, this would mean that only one beamlet is processing the complete structure, eliminating the advantage of the multi-beam setup. Thus, this approach is most suitable for structures that are about the same size as the pitch of the beamlets in the beam pattern. Overall, this adaption of the beamlet segmentation algorithm provides an excellent solution for avoiding visible transition for structures with small, discrete structure elements.

4. Real-time part of the control system

To meet the strict timing requirements while triggering the laser and AOMs synchronously to the actual processing position, a FPGA-based control system has been developed. The system communicates with the control software via network and synchronizes with the scanner, laser, and machine control via digital interfaces.

4.1. Hardware

The basis of the control system is a commercially available board from Avnet: UltraZed-EG SOM. Beside the Zynq UltraScale+ MPSoC chip from Xilinx, it features a 2 GB large memory block. If we consider the maximum size of a layer to be 288 MB ((6 mm/1 µm)^2·64 bit), this RAM size is sufficient for the implementation of a ring buffer for storing at least three layers at a time to ensure continuous processing. Its theoretical maximum data access rate of 2400 Mbps is two orders of magnitude larger than the 24.5 Mbps needed for reading a 64-bit bitmask with 400 kHz. Moreover, the UltraZed board offers sufficient digital user I/O-s to flexibly implement the required physical interfaces.

A carrier card for this device has been designed, that implements standard features, such as power supply as well as network, programming, and serial communication interfaces. The rest of the design is kept modular and prepared for use with piggyback boards to realize the required physical interfaces to each of the following system components: scanner, for position sniffing and handshaking, AOM control, laser triggering and synchronization, and handshaking with the machine control. Each piggyback board is galvanically isolated from the carrier. Due to the low voltage and relatively high frequency of the position-sniffing signals coming from the CUA32 controller, this interface was modified to differential transmission. The ready-to-run system with the carrier card and piggyback boards in a machine-integrable housing is shown in Fig. 20.

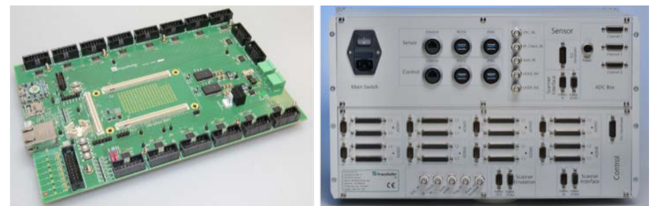

Fig. 20 Hardware of the control system. Left: FPGA carrier card with 16 interfaces for piggyback boards. Right: control system housing with physical connections to AOMs, laser, machine, scanner, and optional sensors.

4.2. Software

Due to its execution on a SoC, the software is divided into two parts: the FPGA part, also called programmable logic (PL) and the processing system (PS) part. The PL part is responsible for connecting and controlling the real-time hardware interfaces. The PS part handles the network communication with the control software. Internally, the two parts communicate via on-board RAM, which is directly connected to PS and addressed by the PL via the high-performance (HP) full power-domain (FPD) ports.

The scanner position sniffing interface is based on a serial protocol with the internal frequency of 10 MHz and the 20-bit data rate of 100 kHz. To fully exploit the hardware performance, i.e. the maximum AOM trigger frequency of 400 kHz, an interpolation of the scanner position has been implemented. The interpolation assumes a constant scanner acceleration and divides one 10 µs cycle into four equal 2.5 µs long cycles.

All AOMs are triggered for 250 ns at a rate of 400 kHz, to establish a constant temperature by heat accumulation. A delay of 2 µs ensures that only the AOMs of the beamlets that are switched on are triggered synchronously to the laser. For protection against overheating, the AOM's duty cycle is monitored internally, an excess of 10% is signaled by a LED on the housing of the system.

For synchronous triggering of AOMs and the laser, the difference in their reaction delays must be considered. Here, it is approx. 4 µs, but since it varies greatly in different systems, it is implemented as an adjustable parameter. To

maintain a once set delay after power-cycling the system, it is stored in the non-volatile memory (EEPROM) available on the FPGA board. During an adjustment phase, this delay can be changed with 10 ns resolution via pushbuttons and the current value is reported back to user software via network, from where additionally its update in the non-volatile memory can be triggered. The information exchange between PS and PL necessary for these actions takes place directly via an AXI-bus (Advanced eXtensible Interface).

The layer processing procedure starts at the PS side when it receives new data from the user software and buffers it in the RAM. Beside the layer information, PS and PL exchange status information via defined registers in RAM. This is how PS signals the offset address of each layer to the PL, to ensure the read address of each pixel can be correctly computed. PL uses scanner handshake inputs to signalize the data has been received and it is ready for processing. Considering handshaking inputs about the processing status, and the interpolated scanner position, the FPGA computes the address of the 64-bit bitmask for each pixel in the RAM. This happens for every scanner coordinate pair at a rate of 400 kHz.

5. Conclusion & outlook

The presented MultiFlex approach enables large-scale, flexible multibeam laser micro-processing with ultra-short pulse lasers in the kilowatt average output power class. This will significantly increase productivity up to a factor of 64, compared to a scanner based single beam process. Hence, it will make laser micro-structuring cost-effective and enable business cases for new fields of applications.

This paper presents considerations regarding the control system for a complex, multibeam-setup utilizing AOMs to realize a fully automated and flexible beam pattern, where every beamlet can be controlled individually. The challenge of calibration of a large-area beam pattern with an edge length of ~40 mm is presented and a system for automated distortion measurement and compensation for every single beamlet is described.

For data processing, an algorithm to generate processing data for the multi-beam system from a bitmap image is presented, including optimizations to address the issue of visible transitions at patch boarders. Here, we present either a shifting of boarders between processing layers or “smart segmentation” to avoid cutting off structural elements at these boarders altogether.

To ultimately complete the software design, an FPGA-based system for synchronous, actual processing position-dependent triggering of laser and AOMs has been developed and is here thoroughly described.

The next steps in the MultiFlex project include the implementation of a runtime & duty cycle simulation for the multi-beam system. This allows further development of the bitmap segmentation algorithm as well as new job generation algorithms to be quickly evaluated for the effectiveness in increasing productivity compared to single beam processing.

Acknowledgements

This work was funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825201.

References

[1] F. Bauer, A. Michalowski, T. Kiedrowski, and S. Nolte: Opt. Express, 23, (2015) 1035.

[2] A. Brenner, M. Zecherle, S. Verpoort, K. Schuster, C. Schnitzler, M. Kogel-Hollacher, M. Reisacher, and B. Nohn: J. Laser Appl., 32, (2020) 12018.

[3] S. Bruening and G. Hennig: Proc. SPIE, Vol. 9351, (2015) 935112.

[4] S. L. Campanelli, A. D. Ludovico, C. Bonserio, P. Cavalluzzi, and M. Cinquepalmi: J. Mater. Process. Technol., 191, (2007) 220.

[5] K. Sugioka and Y. Cheng: Light Sci. Appl., 3, (2014) e149-e149.

[6] B. N. Chichkov, C. Momma, S. Nolte, F. Alvensleben, and A. Tünnermann: Appl. Phys. A, 63, (1996) 109.

[7] P. V. Petkov, S. S. Dimov, R. M. Minev, and D. T. Pham: Proc. Inst. Mech. Eng. B J. Eng. Manuf. , 222, (2008) 35.

[8] P. Russbueldt, T. Mans, J. Weitenberg, H. D. Hoffmann, and R. Poprawe: Optics letters, 35, (2010) 4169.

[9] H. Stark, J. Buldt, M. Müller, A. Klenke, and J. Limpert: Opt. Lett., 46, (2021) 969.

[10] G. Raciukaitis: JLMN, 4, (2009) 186.

[11] K. van der Straeten, O. Nottrodt, M. Zuric, A. Olowinsky, P. Abels, and A. Gillner: Procedia CIRP, 74, (2018) 491.

[12] T. Barthels, M. Reininghaus, and H. Westergeling: Proc. SPIE, Vol. 11107, (2019) 111070K.

[13] L. Büsing: "Optische Systeme für die hochpräzise, scannerbasierte Multistrahlbearbeitung mit ultrakurzen Laserpulsen", Dissertation (RWTH Aachen, 2016).

[14] T. Barthels and M. Reininghaus: Proc. SPIE, Vol. 10744, (2018) 107440B.

[15] A. Meyer, J. Finger, O. Nottrodt, and M. Jüngst: Proc. Lasers Manuf., (2019) 276.

[16] S. Bruening, K. Du, M. Jarczynski, and A. Gillner: J. Laser Appl., 32, (2020) 12003.

[17] J. Finger and M. Hesker: J. Phys. Photonics, 3, (2021) 21004.

[18] S. Dirks, A. Meyer, and M. Kröger: "OpenVectorFormat", <https://github.com/Digital-Production-Aachen/OpenVectorFormat>.

[19] Scanlab AG: "RTC5 Manual" (Puchheim, 2015) 124.

[20] O. Hofmann, J. Stollenwerk, and P. Loosen: Journal of Laser Applications, 32, (2020) 12005.

(Received: June 25, 2021, Accepted: September 23, 2021)