

Nyílt forráskódú mozgás- és deformációelemző szoftver fejlesztése Python környezetben

Development of open-source software for motion and deformation analysis in a Python environment

MÁTRAHÁZI Krisztián Márk MSc hallgató, **Dr. KOSSA Attila** egyetemi docens

Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar, Műszaki Mechanikai Tanszék,
H-1111 Budapest, Műegyetem rkp. 3; Tel.: +36-1463-1369; fax: +36-1463-3471
matrahazik@edu.bme.hu, kossa@mm.bme.hu

Abstract

This study presents a custom-developed software designed for motion analysis using high-resolution video recordings. The program is based on digital image correlation (DIC), through which the displacement field is generated within the region of interest. Strains are computed automatically, with the optional calculation of the stress field. The accuracy of the motion tracking algorithm can be assessed using intermediate text files and virtual test videos, allowing for thoughtful parameter refinement. Through multiple test cases, the authors demonstrate that the program can accurately and efficiently analyze planar deformations.

Keywords: motion analysis, DIC, digital image correlation, open-source, Python

Kivonat

Jelen tanulmány egy saját készítésű szoftvert mutat be, amely nagyfelbontású videófelvevételek mozgáselemzését hajtja végre. A program a digitális képkorrelációra (DIC) épül, amellyel előállítja az elmozdulásmezőt a felvétel vizsgált területén. A szoftver automatikusan számítja az alakváltozásokat, opcionálisan a feszültségmezőt is. A mozgáskövető algoritmus hibáit közbeni szövegfájlokkal és virtuális tesztvideókkal is lehet ellenőrizni, ezek alapján a paramétereit tudatosan finomíthatók. A szerzők több mintapélda bemutatásával igazolják, hogy a kifejlesztett program képes síkbeli alakváltozások pontos és hatékony elemzésére.

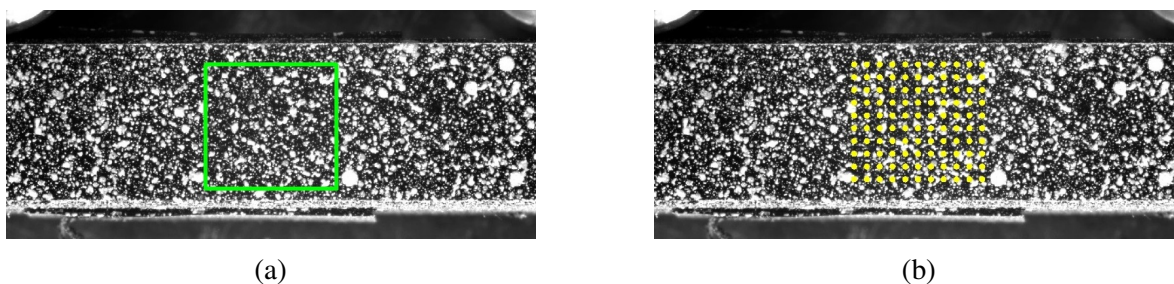
Kulcsszavak: mozgáselemzés, DIC, digitális képkorreláció, nyílt forráskódú, Python

1. BEVEZETÉS

A mechanikai anyagvizsgálatok területén folyamatosan nő a digitális mozgáselemzéssel elvégzett tesztek szerepe. Az ilyen vizsgálatok nagy előnye, hogy érintés- és roncsolásmentesen lehet ezeket kivitelezni, emellett nagy mennyiségű adatot szolgáltatnak, így pontosabb képet kaphatunk a végbemenő folyamatokról.

Jelen tanulmány célja a saját készítésű, nagyfelbontású felvételek kezelésére alkalmas síkbeli mozgáselemző szoftver bemutatása, amely ingyenesen elérhető képfeldolgozó algoritmusokra épül Python környezetben. Az interneten elérhető szoftveres megoldások (főként dedikált DIC programok) sokszor drágák, és kevés, vagy zérus lehetőséget biztosítanak a háttérben futó kód testreszabására, amely igény azonban kutatási feladatok esetén gyakran felmerül. Ezt szem előtt tartva, a szoftver több, előre beépített opcionális lehetőséggel rendelkezik, így például a merev testszerű mozgások kiszűrése az elmozdulásértékekből, a csomóponti alakváltozások és feszültségek átlagolása az interpolációt megelőzően, de ide tartozik a mozgáskövető algoritmus paramétereinek közvetlen módosítása is.

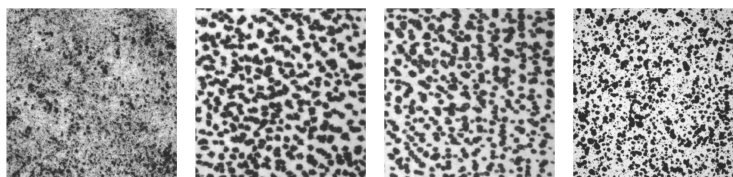
A felhasználó munkáját könnyíti, hogy közvetlenül a felvételen választhatja ki a vizsgálni kívánt területet az 1. (a) ábrának megfelelően, illetve az is, hogy a program eleve videós formában fogadja a bemeneti felvételt, szemben az interneten elérhető megoldások többségével, amelyek különálló képek sorozatát kérik bemenetként. A fejlesztés során fontos szempont volt, hogy a számított mezőváltozókat vizuálisan ráillesszük a bemeneti videó vizsgálatra kijelölt területére, ezzel létrehozva a program vizuális eredményét, szintén videós formában. A szoftver helyességének ellenőrzésére számos közbeni eredményt tartalmazó, futás közben íródo szövegfájl ad lehetőséget. A mozgáskövető algoritmus paramétereinek finomhangolását külön kódrészlet által generált virtuális tesztvideók teszik lehetővé. A tanulmány több mintapéldát is tartalmaz, amelyek igazolják a szoftver képességeit.



1. ábra. A vizsgálandó terület kijelölése (a), illetve az ott elhelyezett, lekötött pontthalmaz (b).

2. A MOZGÁSELEMZÉS ISMERTETÉSE

A digitális mozgáselemzés módszere legtöbbször - csakúgy, mint jelen tanulmányban is - a digitális képkorrelációra épül [1], angolul *digital image correlation*, mozaikszóval DIC. A módszer jelentős hátránya, hogy a vizsgált területnek optikailag inhomogénnek kell lennie a szürkeárnyalatos intenzitásértékek szempontjából. Ha ez a felvételen nem teljesül, akkor a DIC módszer könnyen fals eredményre juthat, hiszen a következő képkockán sok, intenzitás szempontjából hasonló pont mutathat magas korrelációt egy adott referenciaponttal. Gyakorlatban ezt a problémát úgy kezelik, hogy a próbatestet egyfajta pettyes mintázattal látják el, például festékszórás segítségével. Ez hivatott biztosítani a mintadarab kellő optikai inhomogenitását. A 2. ábrán látható minták jól használhatók DIC alkalmazásokra.



2. ábra. Jó mintázatok Lovaas [2] szerint.

A mozgáselemzés a referenciaképen követni kívánt pontok kijelölésével kezdődik. A pont kifejezés jelen esetben nem egy diszkrét pixelt, avagy a mintázat egy pöttyét jelöli, sokkal inkább a fizikai valóságban létező pontra, entitásra érdemes gondolni jelen szöveggörnyezetben. Az 1. (a) ábrán látható területkijelölést elvégezve, a szoftver az előre beállított vízszintes és függőleges osztásközökkel elhelyezi azt a pontthalmazt a téglalap alakú vizsgált területen, amely a futás során a mozgáskövető algoritmus bemeneteként szolgál. Ezen rácspontokat a program egyértelműen jelzi a felhasználó számára, amint azt az 1. (b) ábra mutatja.

A programban a Lucas-Kanade (LK) algoritmus [3] valósítja meg a mozgáskövetést, amely egyszerre n darab pont lekötésére képes. Az eljárás kettő fontos feltételezést tesz: a követett objektum pixeleinek intenzitása nem változik számottevően kettő egymást követő képkocka között, illetve a szomszédos pixelek hasonló pályát követnek. A mozgáskövető algoritmusnak tehát két összehasonlítandó képre, illetve a keresett pontok referenciaképi koordinátaíra van szüksége az 1. táblázatban látható, a programból közvetlenül módosítható egyéb paraméterek mellett. Fontos megemlíteni, hogy a bemeneti videófelvételt külön képkockák sorozatára bontja a szoftver, ezzel eleget téve az LK algoritmus kívánalmainak.

Mindezek alapján, minden betáplált pontot egy észszerű méretű „ablakon” belül keres az algoritmus. A legkisebb négyzetek módszerét alkalmazva, egy iteratív megoldás után kapjuk a keresett pontok koordinátáit a következő képkocka koordináta-rendszerében. Ezen eljárást egy *for* ciklusba ágyazzuk, ahol minden szomszédos képpárra vonatkozó iterációban a korábban megtalált koordináták válnak keresett pontokká. Ily módon a ciklus lefutása után minden vizsgált képkockán ismerjük a követett rácspontok koordinátáit. Ekkor az elmozdulások számítása úgy történik, hogy minden képkockát a legelsőhöz viszonyítunk, majd képezzük az egymásnak megfelelő koordináták különbségét. Így például az első képi elmozdulásmező azonosan nulla.

1. táblázat. A Lucas-Kanade módszer fontosabb paraméterei [4].

Név	Leírás
<i>winSize</i>	Az egyes pontok keresési ablakának mérete pixelben.
<i>criterium.epsilon</i>	Az iteráció leáll a keresési ablak ε -nál kisebb elmozdulása esetén.
<i>criterium.maxCount</i>	Az iteráció leáll a beállított számú lépés után.

Sok felvételen a vizsgálat jellegéből adódóan, esetleg a kamera nem kívánt mozgása miatt könnyen előfordulhatnak merev testszerű elmozdulások és rotációk két képkocka között. Ezek kiszűrése nem létszükséglet az alakváltozási mező számítása, vagyis a numerikus deriválás szempontjából, ennek ellenére egy kiegészítő funkcióként implementáltuk a szoftverbe. Keressük tehát azt az $\mathbf{R} \in \mathbb{R}^{2 \times 2}$ rotációs mátrixot és $\mathbf{t} \in \mathbb{R}^{2 \times 1}$ translációs vektort, amely a legkisebb négyzetek módszerének felhasználásával a legpontosabban átranszformálja a referenciaképi $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ ponthalmaz elemeit a következő képkockához tartozó $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$ ponthalmaz megfelelő pontjaiba, ahol $\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^{2 \times 1}$, $i = 1, 2, \dots, n$. A számítás Sorkine-Hornung és Rabinovich [5] cikkében leírtak alapján történik, ahol a módszer általánosítása és bizonyítása is olvasható. Eszerint minimalizálandó a

$$\sum_{i=1}^n \|(\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i)\|^2 \quad (1)$$

kifejezés. A számítás alapját a szinguláris felbontás adja. Az $\mathbf{R}$ és $\mathbf{t}$ mennyiségek ismeretében előállítható az a $\mathbf{Q}^*$ ponthalmaz, amely kizárólag a merev testszerű mozgás hatását tartalmazza $\mathcal{P}$ -hez képest. Amennyiben az egyes pontok koordinátáit egymás melletti oszlopokba rendezzük, létrehozhatók a $\mathbf{Q}, \mathbf{Q}^* \in \mathbb{R}^{2 \times n}$ mátrixok. A merev testszerű mozgásoktól mentesített elmozdulás-értékek tehát megfogalmazhatók a

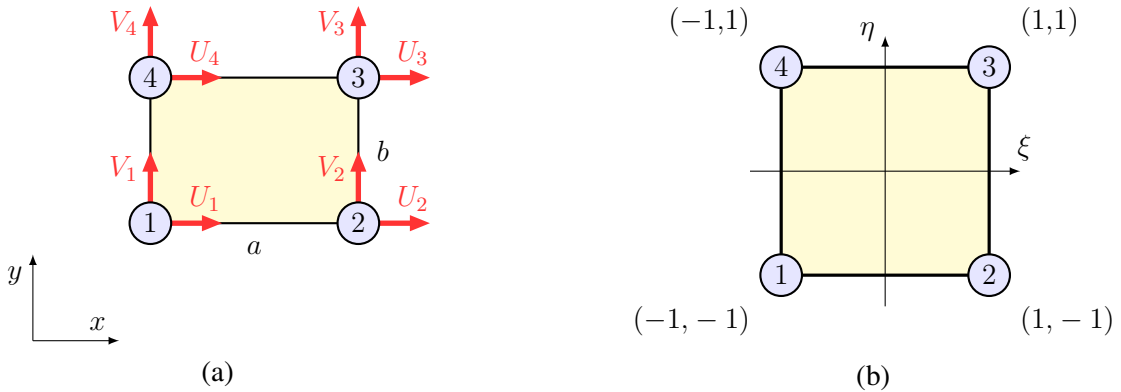
$$\mathbf{D} = \mathbf{Q} - \mathbf{Q}^* \quad (2)$$

mátrixban. Ezt a műveletsorozatot természetesen minden egymást követő képpárra el kell végezni.

3. A MEZŐVÁLTOZÓK SZÁMÍTÁSA, INTERPOLÁCIÓJA

A mozgáskövetés fejezetében a követett rácsponatok időbeli elmozdulás-értékeit előállítottuk az ismertett módszerek egyikével. Tekintve, hogy a teljes vizsgált terület téglalap alakú, amely felbontható az osztásközök okán a és b oldalhosszúságokkal rendelkező kisebb téglalapokra, az interpoláció megvalósítására kézenfekvő választás a végelem módszerből ismert négycsomópontos lineáris négyyszögelem bevezetése. A $\xi - \eta$ lokális koordináta-rendszere ($\xi, \eta \in [-1, 1]$) való áttérést követően a jól ismert bilineáris formafüggvényeket (interpolációs függvényeket) használjuk, amelyek a (3) összefüggésben olvashatók.

$$\begin{aligned} N_1 &= \frac{1}{4}(1 - \xi)(1 - \eta), & N_2 &= \frac{1}{4}(1 + \xi)(1 - \eta), \\ N_3 &= \frac{1}{4}(1 + \xi)(1 + \eta), & N_4 &= \frac{1}{4}(1 - \xi)(1 + \eta). \end{aligned} \quad (3)$$



3. ábra. Koordináta-rendszerek: globális (a), lokális (b).

A 3. ábra összefoglalja egy adott elem globális és lokális koordináta-rendszerbeli reprezentációját, mutatja a vízszintes (U_i) és függőleges (V_i) elmozdulás-komponensek értelmezését, illetve a téglalap alakú elem a és b oldalát. Az izoparametrikus elemdefiníciót használva a közismert $\mathbf{u} = \mathbf{N} \cdot \mathbf{U}$ összefüggéssel írjuk le az elmozdulásmezőt, ahol $\mathbf{N} \in \mathbb{R}^{2 \times 8}$ a formafüggvényeket, $\mathbf{U} \in \mathbb{R}^{8 \times 1}$ pedig a csomóponti elmozdulásokat tartalmazza a megfelelő elrendezésben.

Az alakváltozások számításánál feltételezzük, hogy képről-képre csak kicsi, szemmel nehezen látható alakváltozások lépnek fel. Ebből kiindulva azt is feltehetjük, hogy a kezdetben a és b oldalhosszúságokkal jellemzett téglalap elemek alakja a mozgás során nem torzul. Ezáltal a teljes vizsgált terület is téglalap alakú

marad. Ennek megfelelően, a kis alakváltozásokra érvényes alakváltozási tenzor mátrixa az alábbi módon írható fel az elmozdulásmező segítségével:

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad (4)$$

amelyből a jól ismert $\varepsilon = \mathbf{B} \cdot \mathbf{U}$ felírás vezethető le. Ekkor a $\mathbf{B} \in \mathbb{R}^{3 \times 8}$ mátrix felel a numerikus differenciálásért, hiszen a (3) egyenlet formafüggvényeinek x és y koordináták szerinti parciális deriváltjait tartalmazza. Ezen formafüggvények ξ és η szerinti parciális deriváltjai azonban konstans értékűek, és a globális koordináta-rendszerbe történő leképezés ezen a jelenségen nem sokat torzít abban az esetben, ha az elemek jó közelítéssel téglalap alakúak. Ez azt jelenti, hogy az alakváltozási mező alapvetően nem folytonos, például egy belső csomópontban összekapcsolódó négy elem felől nézve négy különböző ε_x érték adódhat.

A végeselemes szoftverekben gyakori funkció az alakváltozások és a feszültségek csomóponti átlagolása az interpolációt megelőzően, ezzel biztosítva a mezőváltozók folytonosságát. Ez a lehetőség a saját fejlesztésű programban is elérhető. Ekkor az $\varepsilon = \mathbf{N} \cdot \bar{\varepsilon}$ képlet válik aktuálissá, amelyben $\mathbf{N} \in \mathbb{R}^{3 \times 12}$ a formafüggvényeket, $\bar{\varepsilon} \in \mathbb{R}^{12 \times 1}$ pedig a már átlagolt alakváltozásokat tartalmazza a megfelelő elrendezésben.

Az imént kis alakváltozásokat feltételeztünk, ennek megfelelően a feszültségek számításához a Hooke-törvényt alkalmazzuk. Továbbá, egy kamerával készített síkbeli problémát ábrázoló felvétel esetén a legpraktikusabban alkalmazható feltételezés a síkfeszültségi állapot, így célszerű volt a szoftverben ezt a számítási módot implementálni. Mindezt figyelembe véve, a feszültségekre a $\sigma = \mathbf{D} \cdot \varepsilon$ felírás adódik, ahol

$$\mathbf{D} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}. \quad (5)$$

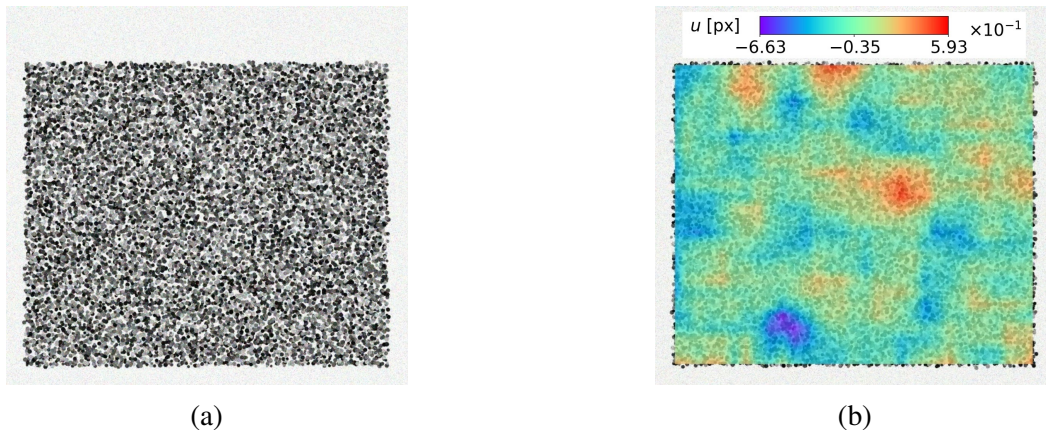
Fontos szempont, hogy a korábban számított ε értékek a feszültségek meghatározása során újrahasznosíthatók. Természetesen a csomóponti átlagolás lehetősége a feszültségeknél is fennáll, ekkor a $\sigma = \mathbf{N} \cdot \bar{\sigma}$ összefüggés alkalmazandó, hasonlóan az alakváltozások esetéhez. A több tengelyű feszültségi állapotok jellemzésére gyakran használt HMM-féle elmélet szerinti (avagy von Mises-féle) egyenértékű feszültség számítására is lehetőség van a programban. Síkfeszültségi állapotra:

$$\sigma_{\text{egy}} = \sqrt{\sigma_x^2 - \sigma_x \sigma_y + \sigma_y^2 + 3\tau_{xy}^2}. \quad (6)$$

4. VIRTUÁLIS TESZTVIDEÓ, MINTAPÉLDÁK

4.1. A virtuális tesztvideók használata

A program fontos kiegészítő eleme egy különálló kódrészlet, mely randomizált tesztvideók generálására képes. Egy ilyen videó egy képkockája, illetve a hozzá tartozó vízszintes irányú elmozdulásmező látható a 4. ábrán. A tesztvideó pöttyös mintázatának több aspektusa, így a pöttyök mérete, elhelyezkedése, szürkeárnyaltos intenzitása is bizonyos határok között véletlenszerű. A limitek módosításával azonban olyan virtuális felvétel készíthető, melynek mintázata jól közelíti a valós mintadarabét. A pöttyök elmozdulása is randomizált, azonban bizonyos osztásközönként (például 20 – 20 pixel vízszintesen és függőlegesen) a pozíciók mentésre kerülnek. Az előző beosztással megegyező hálózással érdemes a tesztvideóra lefuttatni a főprogramot, így az LK algoritmus által nyújtott koordináták közvetlenül összemérhetők a pontosan ismert pozíciókkal. Az egymásnak megfelelő koordináták különbsége felfogható a mozgáskövető algoritmus tévedéseként.

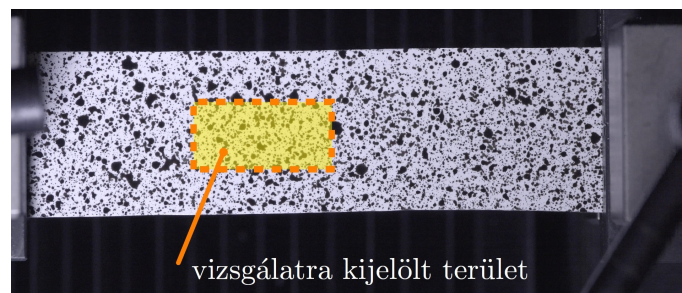


4. ábra. A virtuális tesztvideó egy kiragadott képkockája (a), illetve az ehhez tartozó, főprogram által számított x -irányú elmozdulásmező (b).

Ily módon a virtuális tesztvideók használata erős támpontot biztosít a Lucas-Kanade algoritmus finomhangolásához, hiszen az 1. táblázatban foglalt paraméterek tudatos módosításával a tévedések száma és mértéke hatékonyan csökkenthető. Egyértelműen elkerülendő például, hogy olyan mértékű tévedések alakuljanak ki, amelyek a beállított osztásközt, vagy annak többszöröseit közelítik. Jelen esetben például a 18 ... 22 pixel közötti tévedések nem csak kiugróan magasak, de arról is árulkodnak, hogy az LK algoritmus bizonyos rácspontokat feltehetőleg összetévesztett azok szomszédjaival. Ez természetesen súlyosan hibás eredményre vezetne éles helyzetben. Ennek tükrében a virtuális tesztvideók használata erősen javasolt.

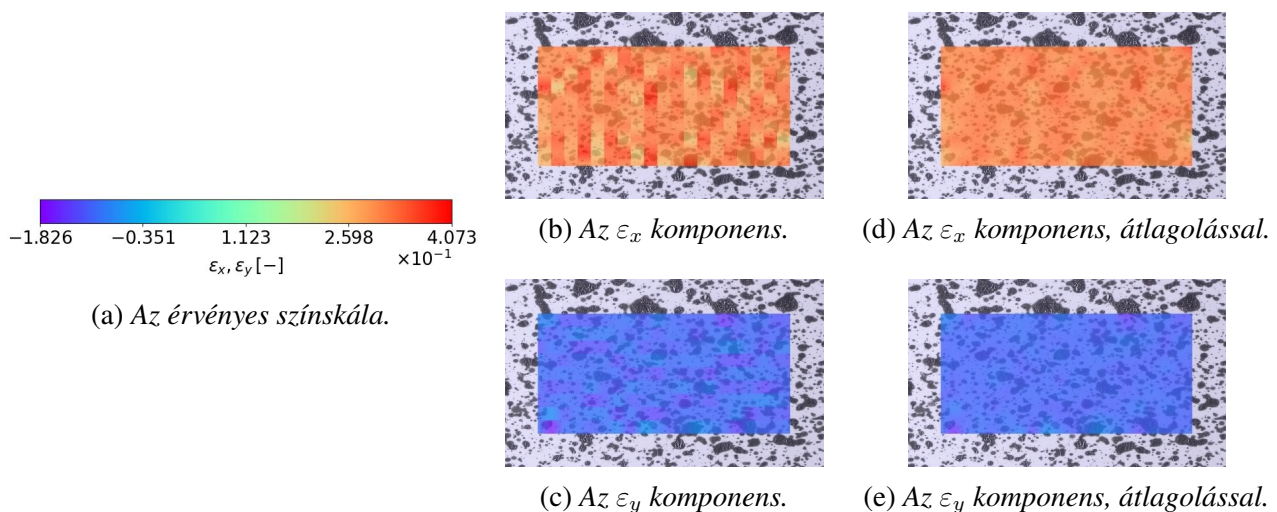
4.2. MC-PET mintadarab egytengelyű húzóvizsgálata

A Műszaki Mechanikai Tanszék laborjában készült a következő felvétel, amelyen egy mikrocellás polietilén-tereftalát (MC-PET) próbatestet vetünk egytengelyű húzás alá. Az 5. ábra mutatja a videófelvétel első képkockáját, illetve a vizsgált területet. A nagyfelbontású (8K) felvételen megfelelő választásnak bizonyult a mindkét irányban 80 pixeles osztásközü pontrács. A mozgáskövető algoritmus paramétereinek megállapításánál figyelembe vettük a festékszórással létrehozott pettyes mintázat sajátosságait.



5. ábra. Az MC-PET mintadarab húzóvizsgálatának első képkockája.

Az alakváltozásokat csomóponti átlagolással és anélkül is meghatároztuk, ahogy azt a 6. ábra mutatja. Az átlagolás nélküli esetben a varakozásainknak megfelelően kirajzolódnak az elemhatárok.

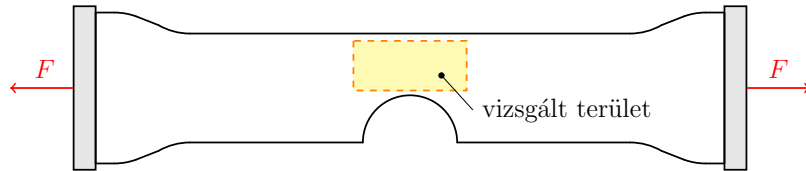


6. ábra. Az MC-PET mintadarab alakváltozásának ε_x és ε_y komponensei a felvétel utolsó képkockáján.

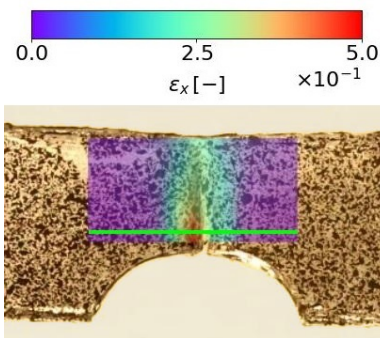
A vizsgált terület egészére vonatkozó átlagos alakváltozás-értékek a következőképpen alakulnak: $\bar{\varepsilon}_x = 0.3212$ és $\bar{\varepsilon}_y = -0.1151$. Az alakváltozások eloszlása jó közelítéssel homogénnek tekinthető, összhangban a Saint-Venant-elvvel, hiszen a terhelési pontok kellően távol esnek a vizsgált területtől. A teljes homogenitástól való kismértékű eltérést okozhatja anyagi inhomogenitás, de a mozgáskövető algoritmus pontatlansága is. A mintadarab Poisson-tényezőjét könnyen számíthatjuk a keresztirányú zsugorodás és a hosszirányú nyúlás hányadosaként. Jelen esetben $\nu = -\bar{\varepsilon}_y / \bar{\varepsilon}_x = 0.3584$.

4.3. Bevágással rendelkező ólom próbatest egytengelyű húzóvizsgálata

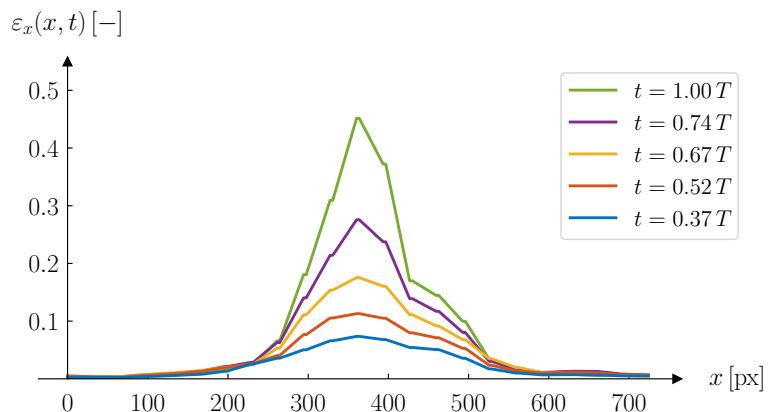
A következő vizsgálat során egy félkör alakú bevágással rendelkező, ólomból készült mintadarabot vetettünk egytengelyű húzás alá. A mérés sematikus elrendezése a 7. (a) ábrán látható. Az alkalmazott pontrács sűrűségét és a mozgáskövető algoritmus paramétereit ismét a pettyes mintázat tulajdonságait figyelembe véve határoztuk meg. A felvétel végére a próbatest képlékeny alakváltozást szenved a legkisebb keresztmetszetenél, a bevágás közelében. A szoftverrel ezt a régiót vizsgáltuk. A 7. (b) ábra mutatja a vízszintes irányú alakváltozást a felvétel utolsó képkockáján.



(a) A mérés sematikus összeállítása.



(b) Az ε_x mező az utolsó képkockán, csomóponti átlagolással.



(c) ε_x értékek az élénkzölddel jelölt vízszintes útvonal mentén.

7. ábra. Az ólom próbatest mérése.

A 7. (b) ábrán elhelyeztünk egy élénkzöld útvonalat, amely mentén a számított ε_x értékeket a 7. (c) ábra jeleníti meg, öt különböző időpillanatban. A felvétel teljes hossza $T = 14$ s. Összességében, a bevágás tetejénél várt kiugró alakváltozás-értékeket jól visszaadja a szoftver, amely ráadásul a képlékeny alakváltozás bekövetkezése előtt kimutatta a kritikus régiót. A bevágástól távolabb eső területeken létrejövő alakváltozások jó közelítéssel homogén eloszlást mutatnak, a Saint-Venant-elv itt is teljesül.

KÖSZÖNETNYILVÁNÍTÁS

Jelen kutatás a Kulturális és Innovációs Minisztérium Nemzeti Kutatási, Fejlesztési és Innovációs Alapból nyújtott támogatásával valósult meg. Pályázati azonosító: FK 142457. A jelen tanulmányban bemutatott tudományos eredmények az MTA Bolyai János Kutatási Ösztöndíj támogatásával készültek.

IRODALMI HIVATKOZÁSOK

- [1] Sutton M. A., Orteu J. J., Schreier H. *Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications*. Springer Science & Business Media, New York, 2009.
- [2] Lovaas N.: *Speckle pattern fundamentals*. Correlated Solution Inc., <https://correlated.kayako.com/article/38-speckle-pattern-fundamentals> (Utolsó letöltés: 2025. 02. 05.).
- [3] Lucas B. D., Kanade T. *An iterative image registration technique with an application to stereo vision*. Proceedings of the 7th International Joint Conference on Artificial intelligence - IJCAI'81, Morgan Kaufmann Publishers Inc., 1981, 4, 674-679.
- [4] Bradski G.: *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, CMP Media, 2000, 120, 122-125.
- [5] Sorkine-Hornung O., Rabinovich M. *Least-squares rigid motion using svd*. Computing, Springer, 2017, 1(1), 1-5.